

AD-A071 395

SYSTEMS TECHNOLOGY INC HAWTHORNE CALIF
MANNED ENGINEERING FLIGHT SIMULATION VALIDATION. PART II. SOFTW--ETC(U)
FEB 79 S A RIEDEL
F33615-77-C-2065

UNCLASSIFIED

STI-TR-1110-2

AFFDL-TR-78-192-PT-2

NL

| OF |

AD
A071395



A071395

LEVEL *III*

A071394

2

AFFDL-TR-78-192
PART II

**MANNED ENGINEERING FLIGHT SIMULATION VALIDATION
PART II: SOFTWARE USER'S GUIDE**

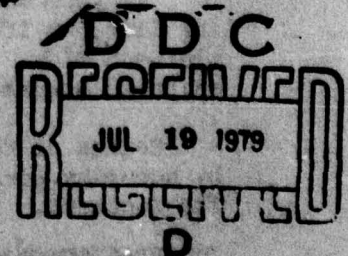
SYSTEMS TECHNOLOGY, INC.
13766 So. Hawthorne Boulevard
Hawthorne, California 90250

February 1979

TECHNICAL REPORT AFFDL-TR-78-192, PART II
Report for Period August 1977 to November 1978

Approved for public release; distribution unlimited

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433



79 07 17 045

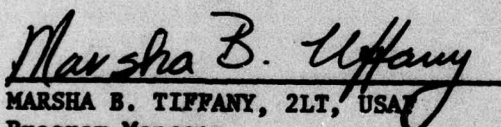
DDC FILE COPY


NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

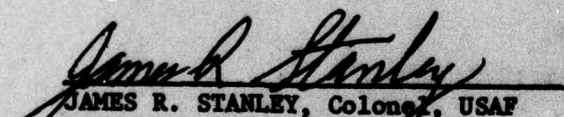
This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.


MARSHA B. TIFFANY, 2LT, USAF
Program Manager


PAUL E. BLATT, Chief
Control Synthesis Branch
Flight Control Division

FOR THE COMMANDER


JAMES R. STANLEY, Colonel, USAF
Chief, Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify _____, W-PAFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFFDL TR-78-192 Pt 2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MANNED ENGINEERING FLIGHT SIMULATION VALIDATION PART II SOFTWARE USER'S GUIDE		5. TYPE OF REPORT & PERIOD COVERED Final Report August 1977-November 1978
6. AUTHOR(s) Susan A. Riedel		7. PERFORMING ORG. REPORT NUMBER TR-1110-2
8. PERFORMING ORGANIZATION NAME AND ADDRESS Systems Technology, Inc. 13766 S. Hawthorne Blvd. Hawthorne, CA 90250		9. CONTRACT OR GRANT NUMBER(s) F33615-77-C-2065
10. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson Air Force Base, OH 45433		11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 54p.		13. REPORT DATE Feb 1979
14. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. SUPPLEMENTARY NOTES		
18. KEY WORDS (Continue on reverse side if necessary and identify by block number) Simulation, drive logic evaluation software, nonlinear washout software, scenario simulation software		
19. ABSTRACT (Continue on reverse side if necessary and identify by block number) The goal of this research is to establish an orderly, relatively simple method for optimizing the presentation of motion cues in moving base simulations. This is accomplished via choice of the drive logic parameters for a given simulator and flying task. The method developed here is based upon use of motion fidelity criteria. It has been applied to optimize the LAMARS motion base drive logic parameters for an air-to-ground scenario. The (continued)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

340 425

B

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

→ methods and procedures can be applied for any given motion base and any desired scenario.

This report describes the FORTRAN computer program used for evaluating and optimizing motion base drive logic. ←

Part I of this report describes the theoretical basis for this computer program.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This report was prepared by Systems Technology, Inc., Hawthorne, California, under United States Air Force Contract F33615-77-C-2065. The program was administered by the Systems Dynamics Branch, Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. The Air Force project engineers were successively John J. Bankovskis and Lt. Marsha B. Tiffany of AFFDL/FGD.

The contract work was performed during the period August 1977 to November 1978. The draft of this report was submitted in September 1978.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	
Unannounced	
Justification _____	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
I. INTRODUCTION	1
II. SOFTWARE OVERVIEW AND SUBROUTINE DESCRIPTIONS	2
Mainline	2
ERRMES	2
OUTR.	3
GPDPLT	3
SENGEN	3
DATGEN	4
READAT	4
HQDT4	5
ROLSWA	5
PITSUR	6
YAWHEV	6
THRES	6
THRESD	6
III. INPUT AND OUTPUT.	31
Mainline	31
GPDPLT	32
SENGEN	32
ROLSWA, PITSUR, YAWHEV.	33
IV. A TYPICAL APPLICATION	34
APPENDIX A. Numerical Integration Technique	50
APPENDIX B. Notes on Software Use.	51

SECTION I

INTRODUCTION

This manual describes for the engineering user the FORTRAN IV software used in the Engineering Flight Simulation Validation study, contract F33615-77-C-2065. Section II includes an overview of the programming strategy, and descriptions of each subroutine. Section III describes the form of the input and output to this set of programs and presents sample input and output files. Section IV provides a complete example for a typical application of this software.

SECTION II

SOFTWARE OVERVIEW AND SUBROUTINE DESCRIPTIONS

The organization of the programs in this software package is divided into two sections — scenario generation, and drive logic stimulation/measures generation. The next sections discuss the FORTRAN subroutines which implement the scenario generation. These are followed by the subroutines for drive logic simulation/measures generation.

MAINLINE

The purpose of the scenario generation mainline routine is to set up an array of variables to be plotted and/or used as inputs to the simulated drive logic. After initialization, the program reads a list of the possible variable names for the particular problem. The program allows for up to 150 variable names — 50 states (Z), 50 derivatives of the states (ZD) and 50 auxiliary variables (XAUX). Thus, a total of 150 variable names must be read into the NVLA array, although many of these names will be blanks. Next, the number of variables to be plotted is read (NV). If NV is greater than 20, an error message is printed and execution is halted. The name of each variable to be plotted is then read, along with its corresponding plotting scale factor, in units/inch (KVLA, SF). Finally, the total length of the time history (TMAX, in seconds) and the time scale factor (TIMSF, in seconds/inch) are read. The program then checks each plotted variable name (KVLA) against the list of all possible variable names (NVLA) — if a plotted variable name is not found in the NVLA list, an error message is printed and execution halts. If all KVLA names are correct, the driving routine for the scenario generation (SENGEN) is called.

ERRMES

This routine is called whenever a fatal error is encountered during execution of any routine in the software package. It prints out a particular error message based on the value of IERR. After ERRMES is called, execution is halted, and the user must correct the error before attempting to rerun the job.

OUTR

This subroutine is called during the scenario generation phase. Its purpose is to buffer the data and interact with the plotting subroutine GPDPLT. After initialization, GPDPLT is called with IO = 0, to plot the axes. The data is then stored in the XOUT array for each variable which was chosen to be plotted in the mainline routine. If the maximum time has been reached ($T = TMAX$) or the IO counter indicates that the XOUT array is full (100 data points for each variable) the accumulated segment for each variable is plotted. Otherwise, the routine returns to its calling routine so that the next data points can be generated and buffered.

GPDPLT

This routine uses CALCOMP plotting routines to plot the time histories for the desired variables. The data is transmitted from OUTR through the DATA array. On initial call, with NPTS = 0, the axes are plotted, five to a page. Thereafter, the data is plotted when GPDPLT is called, in groups of 100 data points per variable. The direction (left or right) in which the data is plotted is dependent upon the position of the pen so as to minimize pen movement. When OUTR sets the switch IEND in GPDPLT's argument list to (1), it signals to GPDPLT that the data in the DATA array is the last to be plotted. When this data has been plotted a label is appended to the time history, the axes are labeled with variable name and scale factor, and the time scale is drawn. Control returns to OUTR. For additional information on the CALCOMP subroutines, refer to the CALCOMP Plotter Manual.

SENGEN

This subroutine is the driving routine for the scenario generation. After initialization, the scenario number is read (see Section III for scenario number definitions). Next, the number of discrete data points for the particular scenario is read, and then the arrays of maximum times, roll accelerations, pitch accelerations and derivative of inertial acceleration which define the inertial characteristics of the trajectory are input

(TMAX, PD, QD, VTDD). The data is generated at 0.1 second intervals. Within each interval, DATGEN is called to generate the remaining scenario data, the data is written to a file for use as a forcing function by the drive logic simulation routine, the units of the data are changed from radians and g's to degrees and ft for plotting purposes, and OUTR is called to buffer the data for plotting. This process continues until the maximum time (TMAX) is reached. Control is returned to the mainline routine.

DATGEN

This routine implements the kinematic equations which generate the variables for the scenario. The initial conditions are determined based on the scenario number. The states and state derivatives (ZO, ZDO) are updated. Then, a subroutine is called, based upon the scenario number, which selects the correct discrete value of roll acceleration, pitch acceleration and derivative of inertial acceleration (PD, QD, VTDD) based on the current time T. (In this package, the subroutines called are HQDT4 for the HQDT4 scenario and READAT otherwise). The kinematic equations are then computed to generate state derivatives (ZD) as well as a_x and a_z [OT(1) and OT(2)]. In addition, the variables (corresponding to actual flight) used as forcing functions by the drive logic simulation are computed (PA, QA, RA, UPOD, VPOD, WPOD). The state derivatives are then "integrated," using a second-order Adams-Bashforth numerical approximation, (see Appendix) to arrive at the states (Z). Whenever an exact algebraic expression is available for computing any state, it is used following the numerical approximation which it then supersedes. Control returns to DATGEN.

READAT

This routine is used for scenario generation whenever the inputs to the kinematic equations (PD, QD, VTDD) can be expressed as discrete values. The purpose of this routine is to compare the current time, T, with the current TMAX value — if T is less than TMAX, the current PD, QD and VTDD values are used. Otherwise, the counter INDEX is incremented and the next discrete value set is used. Control returns to DATGEN.

HQDT4

This subroutine computes the PD, QD and VTDD values from equations, depending on the time T. These equations are used because, for this scenario, continuous input values in distinction to discrete values are pertinent. After the PD and QD values have been determined in degrees/sec², they are converted to radians/sec². Control returns to DATGEN.

ROLSWA

This routine implements the (LAMARS) drive logic-(human pilot) sensory apparatus simulation for the roll and sway axes. In addition, it computes various performance measures, stores them on file and prints them on a line printer. The routine reads the data file output from a scenario generation (QA, PA, RA, UPOD, VPOD, WPOD) and this data is used as input to the drive logic-sensory apparatus simulation. The various parameters (constants) used in the drive logic simulation are input in the initialization section. Also, a number indicating the type of drive logic (linear, subliminal or Parrish) is read into LOGIC, as well as a device number from which the scenario data are to be read (NDEV). Three passes at the drive logic-sensory apparatus simulation equations are necessary to collect the data for the performance measures. In the first pass, the angular velocity axis gain is set to zero. In the second pass, the specific force axis gain is set to zero. The third pass simulates the drive logic using the input data values for both the angular velocity and specific force axis gains. After the state derivatives (ZD) are computed, the values to be used later for the performance measures calculations are accumulated in the AXIS array. AXIS is dimensioned to allow for two axes, 24 measures and eight different scenarios or parameter sets. The state derivatives are "integrated" using a second-order Adams-Bashforth numerical approximation (see Appendix) to obtain the states (Z). This process continues for each pass until the maximum time (TMAX) is reached. The mean of each AXIS value is computed and the desired number of measures columns per page is read. This entire process is repeated from the top of the routine for new parameters or a new scenario (or both) until the desired number of columns is reached

(NCOL = LOOP). The performance measures are then computed (including a change in units from radians to degrees and application of various normalizations), written to a file and printed on the line printer. If additional runs are desired, a 0 is input; otherwise a -1 indicates the end of computation.

PITSUR

This routine simulates the (LAMARS) drive logic and (human pilot) sensory apparatus for the pitch and surge axes, and also computes performance measures. The program logic is identical to that used in the routine ROLSWA, with obvious changes for pitch and surge axes except:

- No provision is made for other than linear drive logic.

YAWHEV

This routine simulates the drive logic and sensory apparatus for the yaw and heave axes. It uses the same logic used by the routines ROLSWA and PITSUR (with the obvious changes to account for the [uncoupled] yaw and heave dynamics) except:

- No provision for nonlinear drive logic, but yaw washout may be first- or second-order.
- Only one pass is needed for drive logic-sensor computations, since yaw and heave axes are uncoupled.

THRES

This routine imposes the complementary function to the indifference threshold which is used in the subliminal (nonlinear) washout scheme. This nonlinear washout is used in the sway axis only so this routine is called by ROLSWA. At its conclusion, it returns control to ROLSWA.

THRES D

This routine is also called by ROLSWA, to impose the partial derivative of the complementary function to the indifference threshold with

respect to its argument. It, too, is part of the subliminal washout scheme. Further explanation of the functions computed by THRES and THRESD can be found in Appendix B of the Engineering Flight Simulation Validation final report.

This concludes the descriptions of the subroutines which comprise this software package. Listings of each subroutine can be found on the pages which follow.

LOCATIONS OF SUBROUTINES

<u>FILENAME</u>	<u>SUBROUTINES CONTAINED</u>
NEWMAN	MAINLINE
SENLIB	SENGEN
	DATGEN
	READAT
	HQDT4
	ERRMES
PLTLIB	OUTR
	GPDPLT
ROLSWA	ROLSWA
PITSUR	PITSUR
YAWHEV	YAWHEV
DLLIB	THRES
	THRESD


```

C
C      MAINLINE FOR PLOTTING
C
C      SETS UP VARIABLE NAMES, SCALE FACTORS
C      SETS UP MAX TIME AND TIME SCALE FACTOR
C      CALLS TIME HISTORY GENERATION SUBROUTINE
C
      DIMENSION NVLA(150)
      COMMON/OUTPUT/NV,KVLA(21),SF(21),KV(20)
      COMMON/IO/IIN,IOUT,ITERM,IPLOT,IMEAS1,IMEAS2
C
C      INITIALIZE
C
      DATA (NVLA(I),I=1,150) /150*1H /
      DATA (KVLA(I),I=1,21),(SF(I),I=1,21) /21*1H ,21*0.0/
      DATA (KV(I),I=1,20),NV /21*0/
      DATA IIN,IOUT,ITERM,IPLOT,IMEAS1,IMEAS2 /20,21,5,5,22,23/
C
C      INPUT LIST OF POSSIBLE VARIABLE NAMES
C
      DO 20 I=1,25
      MIN=6*(I-1)+1
      MAX=6*I
      READ(IIN,10) (NVLA(J),J=MIN,MAX)
      FORMAT(5(A5,1X),A5)
20    CONTINUE
C
C      READ NUMBER OF VARIABLES, VARIABLE NAMES,
C      SCALE FACTORS, MAX TIME, TIME SCALE FACTORS
C
      READ(IIN,30) NV
      FORMAT(I2)
      IF(NV.LE.20) GO TO 40
      CALL ERRMES(1)
      GO TO 1000
40    DO 50 I=1,NV
50    READ(IIN,60) KVLA(I),SF(I)
60    FORMAT(A5,1X,F10.4)
      READ(IIN,70) TMAXT,TIMSF
70    FORMAT(2F10.4)
      SF(NV+1)=TIMSF
C
C      CHECK NAMES
C
      DO 100 I=1,NV
      DO 80 J=1,150
      IF(KVLA(I).EQ.NVLA(J)) GO TO 90
80    CONTINUE
      CALL ERRMES(2)
      GO TO 1000
90    KV(I)=J
100   CONTINUE
C
C      RUN TIME HISTORIES
C
      CALL SENGEN(TMAXT)
C
C      THE END
C
1000  STOP
      ,END

```


SUBROUTINE SENGEN(TMAX)

C

MAIN ROUTINE TO GENERATE SCENARIO

C

DIMENSION Z(9),ZD(9),ZO(9),ZDO(9),Y(9),YD(9),XAUX(3)

COMMON/SCENE/ NSCEN

COMMON/COMDAT/ NDISC,TMAX(50),P(50),Q(50),VT(50)

COMMON/IO/ IIN,IOUT,ITERM,IPLQT,IMEAS1,IMEAS2

C

INITIALIZE

C

DATA (Y(I),I=1,9),(YD(I),I=1,9),(XAUX(I),I=1,3) /21*0.0/

DATA N,IT,IDT,INDEX,DT /9,0,1,1,1/

C

READ SCENARIO NUMBER AND DISCRETE DATA VALUES

C

READ(IIN,10) NSCEN

10

FORMAT(I2)

IF(NSCEN.EQ.3) GO TO 30

READ(IIN,10) NDISC

DO 15 I=1,NDISC

15

READ(IIN,20) TMAX(I),P(I),Q(I),VT(I)

20

FORMAT(4F10.4)

C

CALL KINEMATICS SUBROUTINE, WRITE DATA

C

T=IT/10.

30

CALL DATGEN(T,DT,INDEX,PA,QA,RA,UPOD,VPOD,WPOD,Z,ZO,ZD,ZDO)

WRITE(IOUT,40) PA,QA,RA,UPOD,VPOD,WPOD

40

FORMAT(6(F11.5,1X))

C

CHANGE UNITS, CALL PLOTTING ROUTINE

C

DO 50 J=1,4

Y(J)=Z(J)*57.3

50

YD(J)=ZD(J)*57.3

Y(9)=Z(9)*57.3

YD(9)=ZD(9)*57.3

DO 60 J=7,8

Y(J)=Z(J)

60

YD(J)=ZD(J)

XAUX(1)=UPOD/32.2

XAUX(2)=VPOD/32.2

XAUX(3)=WPOD/32.2

CALL OUTR(T,Y,YD,N,XAUX,TMAX,IENT,IOCNT)

C

IF(T.GT.TMAX) GO TO 70

IT=IT+IDT

GO TO 30

70

RETURN

END

C

SUBROUTINE DATGEN(T,DT,INDEX,PA,QA,RA,UPOD,VPOD,WPOD,Z,ZO,ZD,ZDO)

C

IMPLEMENTS KINEMATIC EQUATIONS

C


```

C      COMMON/SCENE/ NSCEN
      DIMENSION Z(9),ZD(9),ZO(9),ZDO(9),OT(2),ISCEN(3),ZI(9)
      DATA OT(1),OT(2) /2*0.0/

C      G=32.2
      XLX=17.67
      XLZ=-3.1

C      IF(T.NE.0.0) GO TO 50

C      INITIAL CONDITIONS
C
C      DO 40 J=1,9
      ZI(J)=0.0
      Z(J)=0.0
      ZO(J)=0.0
      ZDO(J)=0.0
40     ZD(J)=0.0
      AO=.07
      IF(NSCEN.GT.4) ZI(5)=.07
      IF(NSCEN.EQ.4) ZI(7)=420.
      IF(NSCEN.EQ.4) ZI(8)=-5.
      IF(NSCEN.LT.4) ZI(7)=560.
      IF(NSCEN.GT.4) ZI(7)=800.
      DO 45 J=1,9
45     Z(J)=ZI(J)
C
C      50     DELT2=DT/2.
      DO 60 J=1,9
      ZO(J)=Z(J)
60     ZDO(J)=ZD(J)
C
C      GET DATA
C
C      IF(NSCEN.NE.3) CALL READAT(T,INDEX,PD,QD,VTDD)
      IF(NSCEN.EQ.3) CALL HQDT4(T,Z,PD,QD,VTDD)
C
C      KINEMATIC EQUATIONS
C
C      70     ZD(1)=PD
      ZD(2)=QD
      IF(NSCEN.LE.2) ZD(2)=G*Z(1)*SIN(Z(4))*(1./COS(Z(4)))**2+1)/Z(7)
      ZD(4)=Z(1)
      ZD(5)=Z(2)*COS(Z(4))-Z(3)*SIN(Z(4))
      ZD(6)=Z(2)*SIN(Z(4))+Z(3)*COS(Z(4))
      ZD(7)=Z(8)
      ZD(8)=VTDD
      IF(NSCEN.GT.4) GO TO 74
      OT(1)=Z(8)+G*Z(5)
      OT(2)=-Z(7)*Z(2)-G*COS(Z(4))
      GO TO 76
74     ZD(9)=Z(2)-G*(1.-COS(Z(4)))/Z(7)-Z(9)*Z(7)/ZI(7)
C
      OT(1)=Z(8)+G*Z(5)+Z(7)*(AO+Z(9))*(Z(2)-ZD(9))
      OT(2)=-Z(7)*Z(2)-G*COS(Z(4))+ZD(9)*Z(7)+Z(8)*(AO+Z(9))
C

```

```

76  PA=Z(1)
    QA=Z(2)
    RA=Z(3)
    UPOD=OT(1)+XLZ*ZD(2)+XLZ*Z(3)*Z(1)-XLX*Z(2)**2-XLX*Z(3)**2
    VPOD=-XLZ*ZD(1)+XLX*ZD(3)+XLZ*Z(3)*Z(2)+XLX*Z(1)*Z(2)
    WPOD=OT(2)-XLX*ZD(2)-XLZ*Z(1)**2-XLZ*Z(2)**2+XLX*Z(1)*Z(3)

C
C      INTEGRATE
C
    DO 80 J=1,9
80  Z(J)=ZD(J)+DELT2*(3.*ZD(J)-ZD0(J))
    Z(3)=G*SIN(Z(4))/Z(7)+Z(1)*(Z(9)+A0)
    IF(NSCEN.LE.4) Z(3)=G*SIN(Z(4))/Z(7)
    IF(NSCEN.NE.3) GO TO 90
    IF(T.GT.6.0.AND.T.LE.16.5)
+   Z(1)=(COS(Z(4))**2/(3.*SIN(Z(4))))
    IF(T.GT.5.0.AND.T.LE.17.0)
+   Z(4)=ACOS(1./(2.+(T-5.0)/3.))

C
90  RETURN
    END

C*****
C*****
SUBROUTINE READAT(T,I,PD,QD,VTDD)

C
C      READ DISCRETE SCENARIO GENERATION VALUES
C
COMMON/COMDAT/ NDISC,TMAX(50),P(50),Q(50),VT(50)
N=NDISC

C
IF(T.GE.TMAX(I)) I=I+1
PD=P(I)
QD=Q(I)
VTDD=VT(I)

C
RETURN
END

C*****
C*****
SUBROUTINE HQDT4(T,Z,PD,QD,VTDD)
DIMENSION Z(8)
DPR = 57.3
G = 32.2
VT = 560.0
XNZD = 1./3.

C
PD = 0.0
QD = 0.0
VTDD = 0.0

C
IF(T.GT.2.0.AND.T.LE.3.0) PD = 30.0
IF(T.GT.4.0.AND.T.LE.5.0) PD = -30.0
IF(T.GT.5.0.AND.T.LE.6.0) PD = 5.5
IF(T.GT.6.0.AND.T.LE.16.5) PD = -DPR*Z(1)*XNZD*COS(Z(4))*
+ (1.+(SIN(Z(4))**2.)/(SIN(Z(4))**2.
IF(T.GT.16.5.AND.T.LE.17.5) PD = -0.57

C
IF(T.GT.2.0.AND.T.LE.5.0)

```


PAGE 1-4

SENLIB

FRI 09-JUN-78 16:57

```
+ QD = DPR*G*Z(1)*SIN(Z(4))*(1.+1./(COS(Z(4)))**2.)/VT
  IF(T.GT.5.5.AND.T.LE.17.0)
+ QD = DPR*G*XNZD*(1.+(COS(Z(4)))**2.)/VT
C
  PD = PD/DPR
  QD = QD/DPR
  RETURN
  END
C*****
C*****
  SUBROUTINE ERRMES(IERR)
C
C      ROUTINE TO PRINT OUT ERROR MESSAGES
C
  COMMON/IO/IIN,IOUT,ITERM,IPL0T,IMEAS1,IMEAS2
  IF(IERR.EQ.1) WRITE(ITERM,10)
  IF(IERR.EQ.2) WRITE(ITERM,20)
10  FORMAT(5X,'NO MORE THAN 20 PLOTTING VARIABLES PERMITTED ')
20  FORMAT(5X,'PLOTTING VARIABLE NAME NOT RECOGNIZED ')
  RETURN
  END
```

PAGE 1-1 PLTLIB FRI 09-JUN-78 17:06

```

C****OUTR-START
      SUBROUTINE OUTR(T,Z,ZD,NDIM,XAUX,TMAXT,IEND,IOCNT)
      DIMENSION Z(1),ZD(1),XAUX(1),XOUT(100,21)
      DIMENSION IDATE(2)
C
C      MODIFIED FOR GPDPLT GLT 19-SEP-1977
C
      COMMON/OUTPUT/ NV,KVLA(21),SF(21),KV(20)
C
C      IF(T.NE.0.)GO TO 30
      IOCNT=0
      IOMAX=100
      NVT=NV+1
C
C      PLOT AXIS
C
      CALL GPDPLT(XOUT,NV,IOCNT,KVLA,SF,TMAXT,IEND)
C
C*****
C      UPDATE COUNTER
C
30    IOCNT=IOCNT+1
C
C      STORE DATA
C
      XOUT(IOCNT,NVT)=T
C
C      DO 35 J=1,NV
      I=KV(J)
      IF(I.GT.50) GO TO 32
      XOUT(IOCNT,J)=Z(I)
      GO TO 35
32    IF(I.GT.100)GO TO 34
      XOUT(IOCNT,J)=ZD(I-50)
      GO TO 35
34    XOUT(IOCNT,J)=XAUX(I-100)
35    CONTINUE
C
C      TIME TO PLOT ?
C
      IF(T.GT.TMAXT)IEND=1
      IF(IEND.NE.0)GO TO 40
      IF(IOCNT.LT.IOMAX)GO TO 100
C
C      PLOT SEGMENT
C
40    CALL GPDPLT(XOUT,NV,IOCNT,KVLA,SF,TMAXT,IEND)
C
C      UPDATE OUTPUT MATRIX AND RESET COUNTER
C
      DO 50 J=1,NVT
50    XOUT(1,J)=XOUT(IOCNT,J)
      IOCNT=1
C

```


PAGE 1-2 PLTLIB FRI 09-JUN-78 17:06

```

100 RETURN
END
C$$$OUTR-END
SUBROUTINE GPDPLT(DATA,NVAR,NPTS,LABELS,SF,TMAXT,IEND)
  DIMENSION DATA(100,21),LABELS(21),SF(21),IDATE(2)

C      GENERAL PURPOSE DATA PLOTTING    GLT 3/1/77
C      MOD FOR XV15                      GLT AUG/77

C      DATA    AN ARRAY OF NPTS OF NVAR VARIABLES TO BE PLOTTED VERSUS
C                THE INDEPENDENT VARIABLE, DATA(I,NVAR+1), I=1,NPTS.
C      LABELS    A CORRESPONDING SET OF 5 CHARACTER VARIABLE NAMES.
C      SF        A CORRESPONDING SET OF SCALE FACTORS (UNITS/INCH).
C      TMAXT     THE ULTIMATE MAXIMUM VALUE OF THE INDEPENDENT VARIABLE,
C                WHICH IS ONLY USED TO DETERMINE THE PAGE WIDTH.
C      IEND = 0  MORE DATA TO COME
C                1 THE LAST SET OF DATA

C      GPDPLT WILL PLOT THE DATA USING A ZETA PLOTTER; 5 VARIABLES TO
C      A PAGE. A PAGE IS 11 INCHES HIGH AND A MULTIPLE OF 8.5 INCHES
C      WIDE. THE WIDTH IS DETERMINED BY TMAXT AND SF(NVAR+1). THE
C      INITIAL PEN POSITION SHOULD BE CENTERED (5.5 INCHES FROM THE TOP)
C      AND 1 INCH TO THE RIGHT OF THE LEFT EDGE.

C      GPDPLT IS STRUCTURED TO BE CALLED SEVERAL TIMES TO COMPLETE A
C      SET OF PLOTS. ON THE FIRST CALL (NPTS = 0), GPDPLT IS INITIALIZED
C      AND AXES ARE PLOTTED. ON SUCCEEDING CALLS (NPTS.NE.0) WHATEVER
C      DATA ARE AVAILABLE (I=1,NPTS) ARE ADDED TO THE PLOTS. ON THE
C      FINAL CALL (IEND = 1) THE LAST SET OF DATA IS PLOTTED; A
C      LABEL IS APPENDED TO THE END OF EACH VARIABLE PLOT; AND THE
C      PAPER IS EJECTED.

C      COMMON/IO/IIN,IOUT,ITERM,IPLLOT,IMEAS1,IMEAS2
C      DIMENSION PDATA(1024)

C      *****
C      IF (NPTS.NE.0)GO TO 200

C      *****
C      INITIALIZE

C      NTV=NVAR+1
C      XIN=DATA(1,NTV)
C      XISF=1.0/SF(NTV)
C      XMAX=TMAXT*XISF
C      PWIDTH = 8.5 * ( 1 + INT(XMAX/8.5) )
C      IF(PWIDTH-XMAX.LT.1.0)PWIDTH=PWIDTH+8.5
C      IDIR = -1

C      *****
C      PLOT AXES

C      CALL PLOTS(PDATA,1024,IPLLOT)
C      DO 50 I =1,NVAR

```

PAGE 1-3 PLTLIB FRI 09-JUN-78 17:06

```

      IVAR=I
      IF(I.GT.5.AND.I.LT.11)IVAR = MINO(16-I,NVAR+6-I)
      IF(I.GT.15.AND.I.LT.21)IVAR = MINO(36-I,NVAR+16-I)
      IM1 = IVAR - 1
      MOD5= MOD(IM1,5)
      YORG = 3.5 - 1.5 * MOD5
      XORG = (IM1/5) * PWIDTH
C
      SCALE = 0.5* SF(IVAR)
      LFRAC = 2 - INT(ALOG10(ABS(SCALE)))
      IF(LFRAC.LT.0)LFFAC = 0
C
      CALL SYMBOL(XORG-0.9,YORG,.14,LABELS(IVAR),0.,5)
C
50  CONTINUE
    GO TO 450
C
C *****
C
C          PLOT DATA
C
C      PLOT PAGES LEFT TO RIGHT IF IDIR = +1
C      PLOT PAGES RIGHT TO LEFT IF IDIR = -1
C
200  DO 400 II=1,NVAR
      I=II
      IF(IDIR.EQ.-1)I=NVAR+1-II
      IVAR=I
      IF(I.GT.5.AND.I.LT.11)IVAR=MINO(16-I,NVAR+6-I)
      IF(I.GT.15.AND.I.LT.21)IVAR=MINO(36-I,NVAR+16-I)
      IM1=IVAR-1
      MOD5=MOD(IM1,5)
      YORG=3.5-1.5*MOD5
      XORG=(IM1/5)*PWIDTH
      LDIR=( 1 - 2 * MOD(MOD5,2) ) * IDIR
C
C -----
C
C          PLOT THE IVAR'TH VARIABLE; FOR LDIR = 1, LEFT TO RIGHT
C          -1, RIGHT TO LEFT
C
      YISF=1./SF(IVAR)
      LO = 1
      IF(LDIR.EQ.-1)LO=NPTS
      X=XORG+(DATA(LO,NTV)-XIN)*XISF
      Y=YORG+DATA(LO,IVAR)*YISF
      IF(Y.GT.5.0)Y=5.0
      IF(Y.LT.-5.0)Y=-5.0
      IF(IEND.NE.0.AND.LDIR.EQ.-1)CALL SYMBOL(X+.1,Y-.05,.14,
+      LABELS(IVAR),0.,5)
      CALL PLOT(X,Y,3)
C
      DO 360 L =2,NPTS
        LO=LO+LDIR
        X=(DATA(LO,NTV)-XIN)*XISF
        Y=YORG+DATA(LO,IVAR)*YISF

```


PAGE 1-4

PLTLIB

FRI 09-JUN-78 17:06

```

      IF(Y.GT.5.0)Y=5.0
      IF(Y.LT.-5.)Y=-5.0
      CALL PLOT(X+XORG,Y,2)
360  CONTINUE
      IF(IEND.NE.0.AND.LDIR.EQ.1)CALL SYMBOL(X+XORG+.1,Y-.05,
+      .14,LABELS(IVAR),0.,5)
C
C
C 400  CONTINUE
C
C      IDIR=-IDIR
C
      IF(IEND.EQ.0)GO TO 450
      DO 420 I =1,NVAR
      IVAR=I
      IF(I.GT.5.AND.I.LT.11)IVAR = MINO(16-I,NVAR+6-I)
      IF(I.GT.15.AND.I.LT.21)IVAR = MINO(36-I,NVAR+16-I)
      IM1 = IVAR - 1
      MOD5= MOD(IM1,5)
      YORG = 3.5 - 1.5 * MOD5
      XORG = (IM1/5) * PWIDTH
C
      SCALE = 0.5* SF(IVAR)
      LFRAC = 2 - INT(ALOG10(ABS(SCALE)))
      IF(LFRAC.LT.0)LFRAC = 0
C
      CALL NUMBER(XORG-0.8,YORG+0.5,.14,SCALE,0.,LFRAC)
      CALL PLOT(XORG,YORG+0.5,3)
      CALL PLOT(XORG,YORG,2)
      CALL PLOT(XORG-0.1,YORG,3)
      CALL PLOT(XORG,YORG,2)
      CALL PLOT(XORG,YORG-0.5,2)
C
      IF(IVAR.NE.NVAR.AND.YORG.NE.-2.5)GO TO 420
C
      CALL AXIS(XORG,-4.,4*TIME,-4,1.,0.,XIN,SF(NTU))
420  CONTINUE
      CALL PLOT( (NVAR/5)*PWIDTH+8.5,0,3)
450  CALL PLOTE
500  RETURN
      END

```

PAGE 1-1 ROLSWA MON 31-JUL-78 12:42

```

      DIMENSION Z(18),ZD(18),ZDO(18),ZO(18),XK(2),SCALE(2)
      DIMENSION AXIS(2,29,8),XMEAS(8,24),ILABEL(24),UNIT(2)
      DIMENSION LUN(2),XAUX(8),Y(18),YU(18),PEAK(2),SCALE2(2),XK2(2)
      DATA IIN,ITERM,IMEAS1,IMEAS2 /20,5,21,22/
      DATA (ILABEL(I),I=1,24) /SHPDLA,SHPD1,SHPD2,SHPD3,
+   SHPD4,SHPD5,SHPSENA,SHPSEN1,SHPSEN2,SHPSEN3,SHPSEN4,
+   SHPSENS,SHPNLA,SHPNL1,SHPNL2,SHPNL3,SHPNL4,SHPNLS,
+   SHGS1,SHGS2,SHGSS,SHPACC,SHPVEL,SHPDIS /
      LOOP=1
C
C      GET SCENARIO FILENAME
C
10    READ(IIN,20) NDEV
20    FORMAT(I2)
C
C      CONSTANTS
C
      READ(IIN,30) TMAX,DT
30    FORMAT(2F9.4)
      DELT2=DT/2.
      READ(IIN,70) GP,WP,W1R,W2R,W3R,W4R,G2R,C2Y
70    FORMAT(8F9.4)
      READ(IIN,76) LOGIC,BP,BK,C2YP
76    FORMAT(I2,3F9.4)
C
C      3 PASSES AT DATA
C
      DO 160 IPASS=1,3
      XKP=GP
      XK2R=G2R
      XKY=1.0
      IF(IPASS.EQ.1) XKP=0.0
      IF(IPASS.EQ.2) XK2R=0.0
C
C      INITIALIZE
C
      DO 80 I=1,18
      Z(I)=0.0
80    ZD(I)=0.0
      Z(18)=XKP
      IT=0
      IDT=1
C
C      DATA COMPUTATION LOOP
C
90    T=IT/10.
      DO 100 I=1,18
      ZO(I)=Z(I)
100   ZDO(I)=ZD(I)
C
      READ(NDEV,110) PA,QA,RA,UPOD,VPOD,WPOD
110   FORMAT(6(F11.5,1X))
      VPOD=VPOD/32.2
      IF(LOGIC.EQ.2) GO TO 112
C
      ZD(1)=XKP*PA-WP*Z(1)
      ZD(2)=ZD(1)-Z(4)
      GO TO 114

```



```

112  Z(18)=AMIN1(Z(18),XKP)+AMAX1(Z(18),0.)-Z(18)
      ZD(1)=Z(18)*PA-WP*Z(1)
      ZD(2)=ZD(1)-Z(4)
      ZD(17)=PA-WP*Z(17)
      ZD(18)=(PA-ZD(1))*ZD(17)-BP*Z(1)*Z(17)-BK*(Z(18)-XKP)
      ZD(18)=AMAX1(ZD(18),-0.5)
114  CONTINUE
      ZD(3)=XK2R*VPOD+Z(2)
      ZD(4)=W4R**2*ZD(3)+W1R*W4R**2*Z(3)-(1.4*W3R+W2R)*Z(4)
      IF(LOGIC.NE.3) ZD(5)=ZD(3)-1.4*C2Y**5*Z(5)-C2Y*Z(6)
      IF(LOGIC.EQ.3) ZD(5)=ZD(3)-THRES(.1,1.4*C2Y**5*Z(5)
+ C2Y*Z(6))*THRES(.1,XK2R*VPOD)-1.4*C2Y**5*Z(5)-C2Y*Z(6)
      ZD(6)=Z(5)
      PAF=ZD(2)
      VPOFD=ZD(5)-Z(2)
      ZD(7)=PA-.2*Z(7)
      ZD(9)=PAF-.2*Z(9)
      ZD(11)=1.5*VPOD+.12*Z(12)-1.7*Z(11)-.3*Z(13)
      ZD(12)=VPOD
      ZD(13)=Z(11)
      ZD(14)=1.5*VPOFD+.12*Z(15)-1.7*Z(14)-.3*Z(16)
      ZD(15)=VPOFD
      ZD(16)=Z(14)
C
      PE=XKP*PA-PAF
      PPE=XKP*Z(8)-Z(10)
      PAI=Z(8)-THRES(.035,Z(8))
      PAFI=Z(10)-THRES(.035,Z(10))
      PIE=XKP*PAI-PAFI
      VED=XKY*(Z(2)+VPOD)-ZD(5)
      VEDO=XK2R*VPOD-VPOFD
      VPED=XK2R*Z(11)-Z(14)
      VPOID=Z(11)-THRES(0.1,Z(11))
      VPOFID=Z(14)-THRES(0.1,Z(14))
      VIED=XK2R*VPOID-VPOFID
C
      PASS 1
C
      IF(IPASS.NE.1) GO TO 120
      AXIS(2,2,LOOP)=AXIS(2,2,LOOP)+VED**2
      AXIS(2,4,LOOP)=AXIS(2,4,LOOP)+(Z(2)+VPOD)*ZD(5)
      AXIS(2,8,LOOP)=AXIS(2,8,LOOP)+VPED**2
      AXIS(2,10,LOOP)=AXIS(2,10,LOOP)+Z(11)*Z(14)
      AXIS(2,13,LOOP)=AXIS(2,13,LOOP)+VIED**2
      AXIS(2,15,LOOP)=AXIS(2,15,LOOP)+VPOID*VPOFID
      AXIS(2,25,LOOP)=AXIS(2,25,LOOP)+VPOD*VPOFD
      AXIS(2,28,LOOP)=AXIS(2,28,LOOP)+VEDO**2
C
      PASS 2
C
      CONTINUE
120  IF(IPASS.NE.2) GO TO 130
      AXIS(1,2,LOOP)=AXIS(1,2,LOOP)+PE**2
      AXIS(1,4,LOOP)=AXIS(1,4,LOOP)+PA*PAF
      AXIS(1,8,LOOP)=AXIS(1,8,LOOP)+PPE**2
      AXIS(1,10,LOOP)=AXIS(1,10,LOOP)+Z(8)*Z(10)
      AXIS(1,13,LOOP)=AXIS(1,13,LOOP)+PIE**2
      AXIS(1,15,LOOP)=AXIS(1,15,LOOP)+PAI*PAFI

```

```

C
C
C      PASS 3
130  CONTINUE
      IF(IPASS.NE.3) GO TO 140
      AXIS(1,1,LOOP)=AXIS(1,1,LOOP)+PA**2
      AXIS(1,3,LOOP)=AXIS(1,3,LOOP)+PE**2
      AXIS(1,5,LOOP)=AXIS(1,5,LOOP)+PA*PAF
      AXIS(1,6,LOOP)=AXIS(1,6,LOOP)+Z(8)**2
      AXIS(1,7,LOOP)=AXIS(1,7,LOOP)+Z(10)**2
      AXIS(1,9,LOOP)=AXIS(1,9,LOOP)+PPE**2
      AXIS(1,11,LOOP)=AXIS(1,11,LOOP)+Z(8)*Z(10)
      AXIS(1,12,LOOP)=AXIS(1,12,LOOP)+PAI**2
      AXIS(1,14,LOOP)=AXIS(1,14,LOOP)+PIE**2
      AXIS(1,16,LOOP)=AXIS(1,16,LOOP)+PAI*PAFI
      AXIS(1,21,LOOP)=AMAX1(AXIS(1,21,LOOP),ABS(ZD(2)*57.3))
      AXIS(1,22,LOOP)=AMAX1(AXIS(1,22,LOOP),ABS(Z(2)*57.3))
      AXIS(1,23,LOOP)=AXIS(1,23,LOOP)+PAF**2
      AXIS(1,24,LOOP)=AXIS(1,24,LOOP)+PAFI**2
C
      AXIS(2,1,LOOP)=AXIS(2,1,LOOP)+(Z(2)+VPOD)**2
      AXIS(2,3,LOOP)=AXIS(2,3,LOOP)+VED**2
      AXIS(2,5,LOOP)=AXIS(2,5,LOOP)+(Z(2)+VPOD)*ZD(5)
      AXIS(2,6,LOOP)=AXIS(2,6,LOOP)+Z(11)**2
      AXIS(2,7,LOOP)=AXIS(2,7,LOOP)+Z(14)**2
      AXIS(2,9,LOOP)=AXIS(2,9,LOOP)+VPED**2
      AXIS(2,11,LOOP)=AXIS(2,11,LOOP)+Z(11)*Z(14)
      AXIS(2,12,LOOP)=AXIS(2,12,LOOP)+VPOID**2
      AXIS(2,14,LOOP)=AXIS(2,14,LOOP)+VIED**2
      AXIS(2,16,LOOP)=AXIS(2,16,LOOP)+VPOID*VPOFID
      AXIS(2,20,LOOP)=AMAX1(AXIS(2,20,LOOP),ABS(ZD(5)*32.2))
      AXIS(2,21,LOOP)=AMAX1(AXIS(2,21,LOOP),ABS(Z(5)*32.2))
      AXIS(2,22,LOOP)=AMAX1(AXIS(2,22,LOOP),ABS(Z(6)*32.2))
      AXIS(2,23,LOOP)=AXIS(2,23,LOOP)+VPOFD**2
      AXIS(2,24,LOOP)=AXIS(2,24,LOOP)+VPOFID**2
      AXIS(2,26,LOOP)=AXIS(2,26,LOOP)+VPOD**2
      AXIS(2,27,LOOP)=AXIS(2,27,LOOP)+VPOD*VPOFD
      AXIS(2,29,LOOP)=AXIS(2,29,LOOP)+VEDO**2
C
C      INTEGRATE
C
140  DO 150 I=1,18
      Z(I)=Z0(I)+DELT2*(3.*ZD(I)-ZD0(I))
      IF(I.EQ.8.OR.I.EQ.10) Z(I)=.37*Z0(I)+.63*ZD(I-1)
150  CONTINUE
      IT=IT+IDT
      IF(T.LT.TMAX) GO TO 90
160  CONTINUE
C
C      COMPUTE MEANS
C
      N=TMAX/DT
      DO 170 I=1,2
      DO 170 J=1,29
      IF(J.GE.20.AND.J.LE.22) GO TO 170
      AXIS(I,J,LOOP)=AXIS(I,J,LOOP)/(N+1)
170  CONTINUE
C

```



```

      IF (LOOP.NE.1) GO TO 200
      READ(IIN,190) NCOL
190    FORMAT(I1)
200    LOOP=LOOP+1
      IF (LOOP.LE.NCOL) GO TO 10

C
C      PRINT MEASURES
C

      XK(1)=XKP
      XK(2)=XK2R
      SCALE(1)=XKP
      SCALE(2)=XK2R
      UNIT(1)=57.3
      UNIT(2)=1.0
      LUN(1)=IMEAS1
      LUN(2)=IMEAS2
      PEAK(1)=XKP
      PEAK(2)=XKP
      SCALE2(1)=XKP
      SCALE2(2)=XKY
      XK2(1)=XKP
      XK2(2)=XKY
      DO 228 I=1,8
      DO 228 J=1,24
228    XMEAS(I,J)=0.0
C

      DO 270 IAX=1,2
      IF (SCALE(IAX).EQ.0.0) SCALE(IAX)=1.0
      DO 230 LOOP=1,NCOL
      XMEAS(LOOP,1)=AXIS(IAX,1,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,2)=AXIS(IAX,2,LOOP)/AXIS(IAX,1,LOOP)/SCALE2(IAX)**2
      XMEAS(LOOP,4)=(XK2(IAX)*AXIS(IAX,1,LOOP)-AXIS(IAX,4,LOOP))
+      /AXIS(IAX,1,LOOP)/SCALE2(IAX)
      XMEAS(LOOP,3)=(AXIS(IAX,3,LOOP)-AXIS(IAX,2,LOOP))
+      /AXIS(IAX,1,LOOP)/SCALE2(IAX)**2
      XMEAS(LOOP,5)=(AXIS(IAX,4,LOOP)-AXIS(IAX,5,LOOP))
+      /AXIS(IAX,1,LOOP)/SCALE2(IAX)
      XMEAS(LOOP,6)=AXIS(IAX,23,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,7)=AXIS(IAX,6,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,12)=AXIS(IAX,7,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,8)=AXIS(IAX,8,LOOP)/AXIS(IAX,6,LOOP)/SCALE(IAX)**2
      XMEAS(LOOP,10)=(XK(IAX)*AXIS(IAX,6,LOOP)-AXIS(IAX,10,LOOP))
+      /AXIS(IAX,6,LOOP)/SCALE(IAX)
      XMEAS(LOOP,9)=(AXIS(IAX,9,LOOP)-AXIS(IAX,8,LOOP))
+      /AXIS(IAX,6,LOOP)/SCALE(IAX)**2
      XMEAS(LOOP,11)=(AXIS(IAX,10,LOOP)-AXIS(IAX,11,LOOP))
+      /AXIS(IAX,6,LOOP)/SCALE(IAX)
      XMEAS(LOOP,13)=AXIS(IAX,12,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,14)=AXIS(IAX,13,LOOP)/AXIS(IAX,12,LOOP)/SCALE(IAX)**2
      XMEAS(LOOP,16)=(XK(IAX)*AXIS(IAX,12,LOOP)-AXIS(IAX,15,LOOP))
+      /AXIS(IAX,12,LOOP)/SCALE(IAX)
      XMEAS(LOOP,15)=(AXIS(IAX,14,LOOP)-AXIS(IAX,13,LOOP))
+      /AXIS(IAX,12,LOOP)/SCALE(IAX)**2
      XMEAS(LOOP,17)=(AXIS(IAX,15,LOOP)-AXIS(IAX,16,LOOP))
+      /AXIS(IAX,12,LOOP)/SCALE(IAX)
      XMEAS(LOOP,18)=AXIS(IAX,24,LOOP)*UNIT(IAX)**2
      XMEAS(LOOP,19)=0.0

```

```

      XMEAS(LOOP,20)=0.0
      XMEAS(LOOP,21)=0.0
      XMEAS(LOOP,22)=AXIS(IAX,20,LOOP)/PEAK(IAX)
      XMEAS(LOOP,23)=AXIS(IAX,21,LOOP)/PEAK(IAX)
      XMEAS(LOOP,24)=AXIS(IAX,22,LOOP)/PEAK(IAX)
      IF(IAX.EQ.1) XMEAS(LOOP,22)=0.0
230  CONTINUE
C
      DO 260 J=1,24
      WRITE(LUN(IAX),240) (XMEAS(NLOOP,J),NLOOP=1,NCOL)
240  FORMAT(8F9.2)
      DO 245 NLOOP=1,NCOL
      TEST=ABS(XMEAS(NLOOP,J))
      IF(TEST.GE.100.0.OR.(TEST.LE.0.00005.AND.TEST.NE.0.0)) GO TO 252
245  CONTINUE
      WRITE(ITERM,250) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
250  FORMAT(1X,A5,1X,8F9.5)
      GO TO 260
252  WRITE(ITERM,254) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
254  FORMAT(1X,A5,1X,8E9.2)
260  CONTINUE
      WRITE(ITERM,265)
265  FORMAT(///)
270  CONTINUE
      DO 275 LOOP=1,NCOL
      XMEAS(LOOP,1)=AXIS(2,26,LOOP)*UNIT(2)**2
      XMEAS(LOOP,2)=AXIS(2,28,LOOP)/AXIS(2,26,LOOP)/SCALE(2)**2
      XMEAS(LOOP,3)=(AXIS(2,29,LOOP)-AXIS(2,28,LOOP))/AXIS(2,26,LOOP)
      + /SCALE(2)**2
      XMEAS(LOOP,4)=(XK(2)*AXIS(2,26,LOOP)-AXIS(2,25,LOOP))/
      + AXIS(2,26,LOOP)/SCALE(2)
275  XMEAS(LOOP,5)=(AXIS(2,25,LOOP)-AXIS(2,27,LOOP))/AXIS(2,26,LOOP)
      + /SCALE(2)
      DO 279 J=1,5
      WRITE(LUN(2),240) (XMEAS(NLOOP,J),NLOOP=1,NCOL)
      DO 276 NLOOP=1,NCOL
      TEST=ABS(XMEAS(NLOOP,J))
      IF(TEST.GE.100.0.OR.(TEST.LE.0.00005.AND.TEST.NE.0.0)) GO TO 277
276  CONTINUE
      WRITE(ITERM,250) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
      GO TO 279
277  WRITE(ITERM,254) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
279  CONTINUE
C
      READ(IIN,290) IANS
290  FORMAT(I3)
      IF(IANS.EQ.-1) GO TO 310
C
      DO 300 I=1,2
      DO 300 J=1,24
      DO 300 K=1,8
300  AXIS(I,J,K)=0.0
      LOOP=1
      GO TO 10
C
310  RETURN
      END

```



```

      DIMENSION Z(14),ZD(14),ZDO(14),ZO(14),XK(2),SCALE(2)
      DIMENSION AXIS(2,24,8),XHEAS(8,24),ILABEL(24),UNIT(2)
      DIMENSION LUN(2),PEAK(2)
      DATA IIN,ITERM,IMEAS1,IMEAS2 /20,5,21,22/
      DATA (ILABEL(I),I=1,24) /5HPDLA,5HPDL1,5HPDL2,5HPDL3,
+ 5HPDL4,5HPDLS,5HPSENA,5HPSEN1,5HPSEN2,5HPSEN3,5HPSEN4,
+ 5HPSENS,5HPNLA,5HPNL1,5HPNL2,5HPNL3,5HPNL4,5HPNLS,
+ 5HGS1,5HGS2,5HGS3,5HPACC,5HPVEL,5HPDIS /
      LOOP=1

C
C      GET SCENARIO FILENAME AND OPEN FILE
C
10  READ(IIN,20) NDEV
20  FORMAT(I2)

C
C      CONSTANTS
C
      READ(IIN,30) TMAX,DT
30  FORMAT(2F9.4)
      DELT2=DT/2.
      READ(IIN,70) G2P,WQ,W3P,W4P
70  FORMAT(4F9.4)

C
C      3 PASSES AT DATA
C
      DO 160 IPASS=1,3
      XKQ=0.5
      XK2P=G2P
      IF(IPASS.EQ.1) XKQ=0.0
      IF(IPASS.EQ.2) XK2P=0.0

C
C      INITIALIZE
C
      DO 80 I=1,14
      Z(I)=0.0
80  ZD(I)=0.0
      IT=0
      IDT=1

C
C      DATA COMPUTATION LOOP
C
90  T=IT/10.
      DO 100 I=1,14
      ZO(I)=Z(I)
100  ZDO(I)=ZD(I)

C
      READ(NDEV,110) PA,QA,RA,UPOD,VPOD,WPOD
110  FORMAT(6(F11.5,1X))
      UPOD=UPOD/32.2

C
      ZD(1)=XKQ*QA-WQ*Z(1)
      ZD(2)=ZD(1)+Z(4)
      ZD(3)=XK2P*UPOD-Z(2)
      ZD(4)=(W4P**2)*ZD(3)-1.4*W3P*Z(4)
      QAF=ZD(2)
      UPOD=Z(2)
      ZD(5)=QA-.2*Z(5)
      ZD(7)=QAF-.2*Z(7)

```

PAGE 1-2

PITSUR

MON 31-JUL-78 12:51

```

ZD(9)=1.5*UPOD+.12*Z(10)-1.7*Z(9)-.3*Z(11)
ZD(10)=UPOD
ZD(11)=Z(9)
ZD(12)=1.5*UPOFD+.12*Z(13)-1.7*Z(12)-.3*Z(14)
ZD(13)=UPOFD
ZD(14)=Z(12)

```

C

```

QE=XKQ*QA-QAF
QPE=XKQ*Z(6)-Z(8)
QAI=Z(6)-THRES(.035,Z(6))
QAFI=Z(8)-THRES(.035,Z(8))
QIE=XKQ*QAI-QAFI
UED=XK2P*UPOD-UPOFD
UPED=XK2P*Z(9)-Z(12)
UPOID=Z(9)-THRES(0.1,Z(9))
UPOFID=Z(12)-THRES(0.1,Z(12))
UIED=XK2P*UPOID-UPOFID

```

C
C
C

PASS 1

```

IF(IPASS.NE.1) GO TO 120
AXIS(2,2,LOOP)=AXIS(2,2,LOOP)+UED**2
AXIS(2,4,LOOP)=AXIS(2,4,LOOP)+UPOD*UPOFD
AXIS(2,8,LOOP)=AXIS(2,8,LOOP)+UPED**2
AXIS(2,10,LOOP)=AXIS(2,10,LOOP)+Z(9)*Z(12)
AXIS(2,13,LOOP)=AXIS(2,13,LOOP)+UIED**2
AXIS(2,15,LOOP)=AXIS(2,15,LOOP)+UPOID*UPOFID

```

C
C
C

PASS 2

120

```

CONTINUE
IF(IPASS.NE.2) GO TO 130
AXIS(1,2,LOOP)=AXIS(1,2,LOOP)+QE**2
AXIS(1,4,LOOP)=AXIS(1,4,LOOP)+QA*QAF
AXIS(1,8,LOOP)=AXIS(1,8,LOOP)+QPE**2
AXIS(1,10,LOOP)=AXIS(1,10,LOOP)+Z(6)*Z(8)
AXIS(1,13,LOOP)=AXIS(1,13,LOOP)+QIE**2
AXIS(1,15,LOOP)=AXIS(1,15,LOOP)+QAI*QAFI

```

C
C
C

PASS 3

130

```

CONTINUE
IF(IPASS.NE.3) GO TO 140
AXIS(1,1,LOOP)=AXIS(1,1,LOOP)+QA**2
AXIS(1,3,LOOP)=AXIS(1,3,LOOP)+QE**2
AXIS(1,5,LOOP)=AXIS(1,5,LOOP)+QA*QAF
AXIS(1,6,LOOP)=AXIS(1,6,LOOP)+Z(6)**2
AXIS(1,7,LOOP)=AXIS(1,7,LOOP)+Z(8)**2
AXIS(1,9,LOOP)=AXIS(1,9,LOOP)+QPE**2
AXIS(1,11,LOOP)=AXIS(1,11,LOOP)+Z(6)*Z(8)
AXIS(1,12,LOOP)=AXIS(1,12,LOOP)+QAI**2
AXIS(1,14,LOOP)=AXIS(1,14,LOOP)+QIE**2
AXIS(1,16,LOOP)=AXIS(1,16,LOOP)+QAI*QAFI
AXIS(1,21,LOOP)=AMAX1(AXIS(1,21,LOOP),ABS(ZD(2)*57.3))
AXIS(1,22,LOOP)=AMAX1(AXIS(1,22,LOOP),ABS(Z(2)*57.3))
AXIS(1,23,LOOP)=AXIS(1,23,LOOP)+QAF**2
AXIS(1,24,LOOP)=AXIS(1,24,LOOP)+QAFI**2

```

C


```

      AXIS(2,1,LOOP)=AXIS(2,1,LOOP)+UPOD**2
      AXIS(2,3,LOOP)=AXIS(2,3,LOOP)+UED**2
      AXIS(2,5,LOOP)=AXIS(2,5,LOOP)+UPOD*UPOFD
      AXIS(2,6,LOOP)=AXIS(2,6,LOOP)+Z(9)**2
      AXIS(2,7,LOOP)=AXIS(2,7,LOOP)+Z(12)**2
      AXIS(2,9,LOOP)=AXIS(2,9,LOOP)+UPED**2
      AXIS(2,11,LOOP)=AXIS(2,11,LOOP)+Z(9)*Z(12)
      AXIS(2,12,LOOP)=AXIS(2,12,LOOP)+UPOID**2
      AXIS(2,14,LOOP)=AXIS(2,14,LOOP)+UIED**2
      AXIS(2,16,LOOP)=AXIS(2,16,LOOP)+UPOID*UPOFID
      AXIS(2,23,LOOP)=AXIS(2,23,LOOP)+UPOFD**2
      AXIS(2,24,LOOP)=AXIS(2,24,LOOP)+UPOFID**2

C
C      INTEGRATE
C
140  DO 150 I=1,14
      Z(I)=ZO(I)+DELT2*(3.*ZD(I)-ZDO(I))
      IF(I.EQ.6.OR.I.EQ.8) Z(I)=.37*ZO(I)+.63*ZD(I-1)
150  CONTINUE
      IT=IT+IDT
      IF(T.LT.TMAX) GO TO 90
160  CONTINUE

C
C      COMPUTE MEANS
C
      N=TMAX/DT
      DO 170 I=1,2
      DO 170 J=1,24
      IF(J.GE.20.AND.J.LE.22) GO TO 170
      AXIS(I,J,LOOP)=AXIS(I,J,LOOP)/(N+1)
170  CONTINUE

C
      IF(LOOP.NE.1) GO TO 200
      READ(IIN,190) NCOL
190  FORMAT(I1)
200  LOOP=LOOP+1
      IF(LOOP.LE.NCOL) GO TO 10

C
C      PRINT MEASURES
C
      XK(1)=XKQ
      XK(2)=XK2P
      SCALE(1)=XKQ
      SCALE(2)=XK2P
      UNIT(1)=57.3
      UNIT(2)=1.0
      LUN(1)=IMEAS1
      LUN(2)=IMEAS2
      DO 228 I=1,8
      PEAK(1)=XKQ
      PEAK(2)=XK2P
      DO 228 J=1,24
228  XMEAS(I,J)=0.0

C
      DO 270 IAX=1,2
      IF(SCALE(IAX).EQ.0.0) SCALE(IAX)=1.0
      IF(PEAK(IAX).EQ.0.0) PEAK(IAX)=1.0
      DO 230 LOOP=1,NCOL

```

```

XMEAS(LOOP,1)=AXIS(IA,1,LOOP)*UNIT(IA)**2
XMEAS(LOOP,2)=AXIS(IA,2,LOOP)/AXIS(IA,1,LOOP)/SCALE(IA)**2
XMEAS(LOOP,4)=(XK(IA)*AXIS(IA,1,LOOP)-AXIS(IA,4,LOOP))
+ /AXIS(IA,1,LOOP)/SCALE(IA)
XMEAS(LOOP,3)=(AXIS(IA,3,LOOP)-AXIS(IA,2,LOOP))
+ /AXIS(IA,1,LOOP)/SCALE(IA)**2
XMEAS(LOOP,5)=(AXIS(IA,4,LOOP)-AXIS(IA,5,LOOP))
+ /AXIS(IA,1,LOOP)/SCALE(IA)
XMEAS(LOOP,6)=AXIS(IA,23,LOOP)*UNIT(IA)**2
XMEAS(LOOP,7)=AXIS(IA,6,LOOP)*UNIT(IA)**2
XMEAS(LOOP,12)=AXIS(IA,7,LOOP)*UNIT(IA)**2
XMEAS(LOOP,8)=AXIS(IA,8,LOOP)/AXIS(IA,6,LOOP)/SCALE(IA)**2
XMEAS(LOOP,10)=(XK(IA)*AXIS(IA,6,LOOP)-AXIS(IA,10,LOOP))
+ /AXIS(IA,6,LOOP)/SCALE(IA)
XMEAS(LOOP,9)=(AXIS(IA,9,LOOP)-AXIS(IA,8,LOOP))
+ /AXIS(IA,6,LOOP)/SCALE(IA)**2
XMEAS(LOOP,11)=(AXIS(IA,10,LOOP)-AXIS(IA,11,LOOP))
+ /AXIS(IA,6,LOOP)/SCALE(IA)
XMEAS(LOOP,13)=AXIS(IA,12,LOOP)*UNIT(IA)**2
XMEAS(LOOP,14)=AXIS(IA,13,LOOP)/AXIS(IA,12,LOOP)/SCALE(IA)**2
XMEAS(LOOP,16)=(XK(IA)*AXIS(IA,12,LOOP)-AXIS(IA,15,LOOP))
+ /AXIS(IA,12,LOOP)/SCALE(IA)
XMEAS(LOOP,15)=(AXIS(IA,14,LOOP)-AXIS(IA,13,LOOP))
+ /AXIS(IA,12,LOOP)/SCALE(IA)**2
XMEAS(LOOP,17)=(AXIS(IA,15,LOOP)-AXIS(IA,16,LOOP))
+ /AXIS(IA,12,LOOP)/SCALE(IA)
XMEAS(LOOP,18)=AXIS(IA,24,LOOP)*UNIT(IA)**2
XMEAS(LOOP,19)=0.0
XMEAS(LOOP,20)=0.0
XMEAS(LOOP,21)=0.0
XMEAS(LOOP,22)=0.0
XMEAS(LOOP,23)=AXIS(IA,21,LOOP)/PEAK(IA)
XMEAS(LOOP,24)=AXIS(IA,22,LOOP)/PEAK(IA)
IF(IA.EQ.2) XMEAS(LOOP,23)=0.0
IF(IA.EQ.2) XMEAS(LOOP,24)=0.0
230 CONTINUE
C
DO 260 J=1,24
WRITE(LUN(IA),240) (XMEAS(NLOOP,J),NLOOP=1,NCOL)
240 FORMAT(8E)
DO 245 NLOOP=1,NCOL
TEST=ABS(XMEAS(NLOOP,J))
IF(TEST.GE.100.0.OR.(TEST.LE.0.00005.AND.TEST.NE.0.0)) GO TO 252
245 CONTINUE
WRITE(ITERM,250) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
250 FORMAT(1X,A5,1X,8F9.5)
GO TO 260
252 WRITE(ITERM,254) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
254 FORMAT(1X,A5,1X,8E9.2)
260 CONTINUE
WRITE(ITERM,265)
265 FORMAT(///)
270 CONTINUE
C
READ(IIN,290) IANS
290 FORMAT(I3)
IF(IANS.EQ.-1) GO TO 310
C
DO 300 I=1,2
DO 300 J=1,24
DO 300 K=1,8
300 AXIS(I,J,K)=0.0
LOOP=1
GO TO 10
C
310 STOP
END

```



```

      DIMENSION Z(19),ZD(19),ZDO(19),XK(2),SCALE(2)
      DIMENSION AXIS(2,24,8),XMEAS(8,24),ILABEL(24),UNIT(2)
      DIMENSION LUN(2),XAUX(8),Y(19),YD(19)
      DATA IIN,ITERM,IMEAS1,IMEAS2 /20,5,21,22/
      DATA (ILABEL(I),I=1,24) /5HPDLA,5HPDL1,5HPDL2,
+ 5HPDL3,5HPDL4,5HPDLS,5HPSENA,5HPSEN1,5HPSEN2,
+ 5HPSEN3,5HPSEN4,5HPSENS,5HPNLA,5HPNL1,5HPNL2,5HPNL3,
+ 5HPNL4,5HPNLS,5HGS1,5HGS2,5HGS3,5HPACC,5HPVEL,
+ 5HPDIS /
      LOOP=1

C
C      GET SCENARIO FILENAME
C
10  READ(IIN,20) NDEV
20  FORMAT(I2)
C
C      CONSTANTS
C
      READ(IIN,30) TMAX,DT
      FORMAT(2F9.4)
      DELT2=DT/2.
      READ(IIN,70) IORDER,WR,C3Z,C4Z,C5Z
      FORMAT(I1,4F9.4)
70
C
C      INITIALIZE
C
      DO 80 I=1,19
      Z(I)=0.0
80  ZD(I)=0.0
      IT=0
      IDT=1

C
C      DATA COMPUTATION LOOP
C
90  T=IT/10.
      DO 100 I=1,19
      ZD(I)=Z(I)
100  ZDO(I)=ZD(I)
      READ(NDEV,110) PA,QA,RA,UPOD,VPOD,WPOD
110  FORMAT(6(F11.5,1X))
      WPOD=WPOD/32.2+1.

C
C      YAW
C
      IF(IORDER.EQ.2) GO TO 120
      ZD(1)=0.5*RA-WR*Z(1)
      GO TO 130
120  ZD(1)=0.5*RA-1.4*WR*Z(1)-WR**2*Z(2)
      ZD(2)=Z(1)
130  RAF=ZD(1)
      ZD(3)=RA-.2*Z(3)
      ZD(5)=RAF-.2*Z(5)
      RE=0.5*RA-RAF
      RPE=0.5*Z(4)-Z(6)
      RAI=Z(4)-THRES(.035,Z(4))
      RAFI=Z(6)-THRES(.035,Z(6))
      RIE=0.5*RAI-RAFI
C

```

C
C

HEAVE

```

ZD(7)=0.15*WPOD-(.05+C3Z+C5Z)*Z(7)-(.05*C5Z+C3Z*C5Z
+ +.05*C3Z+C4Z)*Z(8)-(.05*C3Z*C5Z+C4Z*C5Z+.05*C4Z)
+ *Z(9)-.05*C4Z*C5Z*Z(10)
ZD(8)=Z(7)
ZD(9)=Z(8)
ZD(10)=Z(9)
WPOFD=ZD(7)-1.
ZD(11)=1.5*WPOD+.12*Z(12)-1.7*Z(11)-0.3*Z(13)
ZD(12)=WPOD
ZD(13)=Z(11)
ZD(14)=1.5*(WPOFD+1.)+.12*Z(15)-1.7*Z(14)-0.3*Z(16)
ZD(15)=(WPOFD+1.)
ZD(16)=Z(14)
ZD(17)=ZD(11)+0.2*Z(11)-Z(17)
ZD(18)=ZD(14)+0.2*Z(14)-Z(18)
ZD(19)=0.92*WPOD-Z(19)
WED=0.15*WPOD-(WPOFD+1.)
WPED=0.15*Z(11)-Z(14)
WSED=0.15*Z(17)-Z(18)
WIED=0.15*(Z(17)+Z(19))-Z(18)
WPOFSD=0.15*Z(19)-Z(18)

```

C

```

AXIS(1,1,LOOP)=AXIS(1,1,LOOP)+RA**2
AXIS(1,3,LOOP)=AXIS(1,3,LOOP)+RE**2
AXIS(1,5,LOOP)=AXIS(1,5,LOOP)+RA*RAF
AXIS(1,6,LOOP)=AXIS(1,6,LOOP)+Z(4)**2
AXIS(1,7,LOOP)=AXIS(1,7,LOOP)+Z(6)**2
AXIS(1,9,LOOP)=AXIS(1,9,LOOP)+RPE**2
AXIS(1,11,LOOP)=AXIS(1,11,LOOP)+Z(4)*Z(6)
AXIS(1,12,LOOP)=AXIS(1,12,LOOP)+RAI**2
AXIS(1,14,LOOP)=AXIS(1,14,LOOP)+RIE**2
AXIS(1,16,LOOP)=AXIS(1,16,LOOP)+RAI*RAFI
AXIS(1,21,LOOP)=AMAX1(AXIS(1,21,LOOP),ABS(ZD(1)*57.3))
AXIS(1,22,LOOP)=AMAX1(AXIS(1,22,LOOP),ABS(Z(1)*57.3))
AXIS(1,23,LOOP)=AXIS(1,23,LOOP)+RAF**2
AXIS(1,24,LOOP)=AXIS(1,24,LOOP)+RAFI**2

```

C

```

AXIS(2,1,LOOP)=AXIS(2,1,LOOP)+WPOD**2
AXIS(2,3,LOOP)=AXIS(2,3,LOOP)+WED**2
AXIS(2,5,LOOP)=AXIS(2,5,LOOP)+WPOD*(WPOFD+1.)
AXIS(2,6,LOOP)=AXIS(2,6,LOOP)+Z(11)**2
AXIS(2,7,LOOP)=AXIS(2,7,LOOP)+Z(14)**2
AXIS(2,9,LOOP)=AXIS(2,9,LOOP)+WPED**2
AXIS(2,11,LOOP)=AXIS(2,11,LOOP)+Z(11)*Z(14)
AXIS(2,12,LOOP)=AXIS(2,12,LOOP)+(Z(17)+Z(19))**2
AXIS(2,14,LOOP)=AXIS(2,14,LOOP)+WSED**2
AXIS(2,16,LOOP)=AXIS(2,16,LOOP)+(Z(17)+Z(19))*WPOFSD
AXIS(2,17,LOOP)=AXIS(2,17,LOOP)+Z(18)**2
AXIS(2,18,LOOP)=AXIS(2,18,LOOP)+WIED**2
AXIS(2,19,LOOP)=AXIS(2,19,LOOP)+(Z(17)+Z(19))*Z(18)
AXIS(2,20,LOOP)=AMAX1(AXIS(2,20,LOOP),ABS(ZD(7)*32.2))
AXIS(2,21,LOOP)=AMAX1(AXIS(2,21,LOOP),ABS(Z(7)*32.2))
AXIS(2,22,LOOP)=AMAX1(AXIS(2,22,LOOP),ABS(Z(8)*32.2))
AXIS(2,23,LOOP)=AXIS(2,23,LOOP)+(WPOFD+1.)**2
AXIS(2,24,LOOP)=AXIS(2,24,LOOP)+WPOFSD**2

```

C


```

C      INTEGRATE
C
DO 140 I=1,19
Z(I)=Z0(I)+DELT2*(3.*ZD(I)-ZD0(I))
IF(I.EQ.4.OR.I.EQ.6) Z(I)=.37*Z0(I)+.63*ZD(I-1)
140 CONTINUE
IT=IT+IDT
IF(T.LT.TMAX) GO TO 90

C      COMPUTE MEANS
C
N=TMAX/DT
DO 210 I=1,2
DO 210 J=1,24
IF(J.GE.20.AND.J.LE.22) GO TO 210
AXIS(I,J,LOOP)=AXIS(I,J,LOOP)/(N+1)
210 CONTINUE
C
IF(LOOP.NE.1) GO TO 240
READ(IIN,230) NCOL
230 FORMAT(I1)
240 LOOP=LOOP+1
IF(LOOP.LE.NCOL) GO TO 10

C      PRINT MEASURES
C
XK(1)=0.5
XK(2)=0.15
SCALE(1)=0.5
SCALE(2)=0.15
UNIT(1)=57.3
UNIT(2)=1.0
LUN(1)=21
LUN(2)=22
DO 312 I=1,8
DO 312 J=1,24
312 XMEAS(I,J)=0.0
C
DO 360 IAX=1,2
DO 320 LOOP=1,NCOL
XMEAS(LOOP,1)=AXIS(IAX,1,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,2)=AXIS(IAX,3,LOOP)/AXIS(IAX,1,LOOP)/SCALE(IAX)**2
XMEAS(LOOP,4)=(XK(IAX)*AXIS(IAX,1,LOOP)-AXIS(IAX,5,LOOP))
+ /AXIS(IAX,1,LOOP)/SCALE(IAX)
XMEAS(LOOP,6)=AXIS(IAX,23,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,7)=AXIS(IAX,6,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,8)=AXIS(IAX,9,LOOP)/AXIS(IAX,6,LOOP)/SCALE(IAX)**2
XMEAS(LOOP,10)=(XK(IAX)*AXIS(IAX,6,LOOP)-AXIS(IAX,11,LOOP))
+ /AXIS(IAX,6,LOOP)/SCALE(IAX)
XMEAS(LOOP,12)=AXIS(IAX,7,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,13)=AXIS(IAX,12,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,14)=AXIS(IAX,14,LOOP)/AXIS(IAX,12,LOOP)/SCALE(IAX)**2
XMEAS(LOOP,16)=(XK(IAX)*AXIS(IAX,12,LOOP)-AXIS(IAX,16,LOOP))
+ /AXIS(IAX,12,LOOP)/SCALE(IAX)
XMEAS(LOOP,18)=AXIS(IAX,24,LOOP)*UNIT(IAX)**2
IF(IAX.EQ.1) GO TO 315
XMEAS(LOOP,19)=AXIS(IAX,18,LOOP)/AXIS(IAX,12,LOOP)/SCALE(IAX)**2
XMEAS(LOOP,20)=(XK(IAX)*AXIS(IAX,12,LOOP)-AXIS(IAX,19,LOOP))

```

```
+ /AXIS(IAX,12,LOOP)/SCALE(IAX)
XMEAS(LOOP,21)=AXIS(IAX,17,LOOP)*UNIT(IAX)**2
XMEAS(LOOP,22)=AXIS(IAX,20,LOOP)/XK(IAX)
315 XMEAS(LOOP,23)=AXIS(IAX,21,LOOP)/XK(IAX)
XMEAS(LOOP,24)=AXIS(IAX,22,LOOP)/XK(IAX)
320 CONTINUE
C
DO 350 J=1,24
WRITE(LUN(IAX),325) (XMEAS(NLOOP,J),NLOOP=1,NCOL)
325 FORMAT(8E)
DO 330 NLOOP=1,NCOL
TEST=ABS(XMEAS(NLOOP,J))
IF(TEST.GE.100.0.OR.(TEST.LE.0.00005.AND.TEST.NE.0.0)) GO TO 340
330 CONTINUE
WRITE(ITERM,335) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
335 FORMAT(1X,A5,1X,8F9.5)
GO TO 350
340 WRITE(ITERM,345) ILABEL(J),(XMEAS(NLOOP,J),NLOOP=1,NCOL)
345 FORMAT(1X,A5,1X,8E9.2)
350 CONTINUE
WRITE(ITERM,355)
355 FORMAT(///)
360 CONTINUE
C
READ(IIN,380) IANS
380 FORMAT(I3)
IF(IANS.EQ.-1) GO TO 400
C
DO 390 I=1,2
DO 390 J=1,24
DO 390 K=1,8
390 AXIS(I,J,K)=0.0
LOOP=1
GO TO 10
C
400 RETURN
END
```


PAGE 1-1 DLLIB MON 31-JUL-78 13:06

```
C****THRES-START
      FUNCTION THRES(TH,X)
C
C      IMPLEMENTS A THRESHOLD, OF THE FORM
C       $Y = TH * SIN(X/TH)$ , FOR  $-PI*TH/2 < X < PI*TH/2$ 
C      AND
C       $Y = -TH$  FOR  $X = -PI*TH/2$ 
C       $T=TH$  FOR  $X > PI*TH/2$ 
C
      DPR = 57.29578
      PI = 3.14159
      IF(X.LT.0.0) THRES=-TH
      IF(X.GT.0.0) THRES=TH
      IF(X.GE.-PI*TH/2..AND.X.LE.PI*TH/2.)
      + THRES = TH*SIN(X/TH)
      RETURN
      END
C****THRES-END
C****THRESD-START
      FUNCTION THRESD(TH,X)
C
C      IMPLEMENTS A THRESHOLD OF THE FORM
C       $Y=COS(PI*X/2/TH)$ , FOR  $-TH < X < TH$ 
C      AND
C       $Y=0$  , FOR  $X < -TH$  OR  $X > TH$ 
C
      DPR=57.29578
      PI=3.14159
      THRESD=0.0
      IF(X.GE.-TH.AND.X.LE.TH) THRESD=COS(X/TH)
      RETURN
      END
C****THRESD-END
```

SECTION III

INPUT AND OUTPUT

Most of the input and output involved in this subroutine package is stored on files — the only exceptions to this are the hard copy plots which can be produced, and the listing of measures via the line printer. In this section, each subroutine which requires input or produces output will be discussed in terms of detailed descriptions of that input or output. At the end of this section are included sample listings of input and output.

MAINLINE

For the scenario generation mainline, all of the input data is contained on the logical device IIN. The first input is the list of possible plotting variable names. There must be 150 names input: 50 state names, 50 state derivative names and 50 auxiliary variable names, in that order. Each name is 5 characters long. The file must contain 25 lines, 6 names to a line, separated by 1 comma or 1 blank. All names with fewer than 5 characters must be filled out with blanks. Each state variable name must correspond to its appropriate state — that is, the first state variable name [NVLA (1)] must correspond to the first state [Z(1)]. The same holds true for the state derivatives and auxiliary variables [NVLA(51) corresponds to ZD(1) and NVLA(101) corresponds to XAUX(1)]. If there are fewer than 50 states, state derivatives or auxiliary variables, 5 blanks should comprise the remaining variable names. The next input is the number of variables to be plotted. This is an integer greater than 0 and less than 21. Next are the plotting variable names and their corresponding scale factors. Each variable name and its scale factor must occupy a separate line. The variable name must correspond to a name in the list of possible variable names. The variable name must be 5 characters long, the scale factor is in units per inch and must be a real number, and the two must be separated by a comma or a blank. The final input used by this routine is the maximum time, in seconds and the time scale factor, in seconds per inch. Each is a real number.

GPDPLOT

The output from this subroutine is a set of hard copy plots. Up to 20 plots are permitted, five per page. These plots are output to device IPLOT.

SENGEN

There are several inputs to this subroutine — they should follow directly after the data for the mainline discussed above. The first is the scenario number, NSCEN. It is an integer which uniquely identifies which of the eight existing scenarios is desired. The next line contains an integer whose value is the number of discrete data points needed to describe this scenario (NDISC). A list of the scenarios and their corresponding NSCEN and NDISC values is given below:

<u>Scenario</u>	<u>NSCEN</u>	<u>NDISC</u>
HQDT1	1	5
HQDT2	2	9
HQDT4	3	—
AG	4	40
AA1	5	42
AA2	6	16
AA3	7	8
AA4	8	9

The final input for scenario generation is a list of the discrete values from which the time history of the scenario will be computed. (Note, however, that the HQDT4 scenario does not require the input of NDISC or the list of discrete values. This is because the HQDT4 scenario is described by algebraic equations which are functions of time, not by discrete values). The discrete values include TMAX, PA, QA and VTDD. The TMAX value is the maximum value for which the PD, QD and VTDD values hold — when one or more of these values changes, a new TMAX is needed.

A more complete description of the process used to arrive at the discrete values can be found in the Scenario Generation chapter of the Engineering Flight Simulation Validation final report.

The output from the scenario generation is a file or device IOUT which contains the inputs to the drive logic simulation (PA, QA, RA, UPOD, VPOD, WPOD). Each of these variables is computed at 0.1 second intervals for the length of the scenario (TMAX).

ROLSWA, PITSUR, YAWHEV

The inputs to these routines are all very similar, and will be discussed together. There are actually two sets of inputs — the first is the scenario, which resides on logical unit NDEV. This file was produced by the scenario generation package. The other inputs reside on logical unit IIN. These inputs are described below:

- NDEV - logical unit for scenario generation data
- TMAX,DT - maximum length of scenario, and integration interval
- Constants - (for definition of constants, see Final Report)
- For ROLSWA KP, WP, W1R, W2R, W3R, W4R, K2R, C2Y
Drive logic type (1, 2 or 3), BP, BK, C2YP
- For PITSUR K2P, WQ, W3P, W4P
- For YAWHEV IORDER (1 or 2), WR, C3Z, C4Z, C5Z
- NCOL - number of columns. If more than 1 column of data is desired, all of the preceding information must be supplied for each column.
- IANIS - If more data is to be computed, a 0 is input and all preceding information must be included for the next page of data. Otherwise, a -1 input ends the program.

Outputs from each of the three routines are files containing the angular velocity and specific force axes measures (on devices IMEAS1 and IMEAS2, respectively), as well as a hard copy formatted listing of those measures.

SECTION IV

A TYPICAL APPLICATION

This example presents the input files, output files and hard copy associated with generating the AA4 scenario and, subsequently, the performance measures for two different sets of drive logic parameters when the drive logic forcing function is the AA4 scenario. Only the roll and sway measures are presented. Each section of each file is labeled to indicate what variable or set of variables is listed.

A listing of the discrete values used to generate each of the seven remaining scenarios follows this example.

LIST OF INPUTS AND OUTPUTS FOR SAMPLE RUN (Scenario 8)

TINPUT - Contains all inputs to scenario generation software.

Scenario generation plots for Scenario 8

TOUT - Contains time histories for PA, QA, RA, UPOD, VPOD, WPOD variables. This file is output by the scenario generation software, and is read as input to the drive logic/sensory apparatus simulation software.

DINPUT - Contains all inputs to drive logic/sensory apparatus simulation software except those contained in TOUT

Table of measures generated by drive logic/sensory apparatus simulation, for roll-sway axes.

ROLL - File containing roll axis measures

SWAY - File containing sway axis measures

- NVLA

- KVL, SF

8 - NSCEN
9 - NDISC

- TMAX, PD, QD, VTDD

PAGE 1-1

TOUT

FRI 09-JUN-78 16:47

0.00000	0.00000	0.00000	2.25400	-0.59520	-32.20000
-0.02880	0.00000	-0.00202	2.25375	-0.59520	-32.19640
-0.04800	0.00000	-0.00353	2.25327	-0.59520	-32.18986
-0.06720	0.00000	-0.00511	2.25251	-0.59520	-32.17989
-0.08640	0.00000	-0.00676	2.25142	-0.59520	-32.16627
-0.10560	0.00000	-0.00849	2.24993	-0.59520	-32.14867
-0.12480	0.00000	-0.01030	2.24796	-0.59520	-32.12666
-0.14400	0.00000	-0.01218	2.24538	-0.59520	-32.09963
-0.16320	0.00000	-0.01414	2.24206	-0.59520	-32.06682
-0.18240	0.00000	-0.01618	2.23782	-0.59520	-32.02725
-0.20160	0.00000	-0.01829	2.23243	-0.59520	-31.97978
-0.22080	0.00000	-0.02047	2.22564	-0.59520	-31.92304
-0.24000	0.00000	-0.02273	2.21714	-0.59520	-31.85542
-0.25920	0.00000	-0.02505	2.20658	-0.59520	-31.77509
-0.27840	0.00000	-0.02744	2.19356	-0.59520	-31.67998
-0.29760	0.00000	-0.02989	2.17764	-0.59520	-31.56779
-0.31680	0.00000	-0.03240	2.15830	-0.59520	-31.43595
-0.33600	0.00000	-0.03497	2.13501	-0.59520	-31.28167
-0.35520	0.00000	-0.03758	2.10716	-0.59520	-31.10189
-0.37440	0.00000	-0.04023	2.07412	-0.59520	-30.89335
-0.39360	0.00000	-0.04291	2.03520	0.97340	-30.65256
-0.33690	0.00000	-0.04040	2.01380	0.97340	-30.64447
-0.30550	0.00000	-0.03926	1.97763	0.97340	-30.50544
-0.27410	0.00000	-0.03806	1.93798	0.97340	-30.34492
-0.24270	0.00000	-0.03674	1.89389	0.97340	-30.15900
-0.21130	0.00000	-0.03532	1.84614	0.97340	-29.95352
-0.17990	0.00000	-0.03381	1.79531	0.97340	-29.73301
-0.14850	0.00000	-0.03220	1.74205	0.97340	-29.50218
-0.11710	0.00000	-0.03050	1.68702	0.97340	-29.26575
-0.08570	0.00000	-0.02872	1.63087	0.97340	-29.02830
-0.05430	0.00000	-0.02686	1.57426	0.00000	-28.79429
-0.07000	0.00000	-0.02799	1.51063	0.00000	-28.52980
-0.07000	0.00000	-0.02823	1.45062	0.00000	-28.29978
-0.07000	0.00000	-0.02843	1.38973	0.00000	-28.07338
-0.07000	0.00000	-0.02864	1.32878	0.00000	-27.85677
-0.07000	0.00000	-0.02885	1.26755	0.00000	-27.64781
-0.07000	0.00000	-0.02905	1.20598	0.00000	-27.44570
-0.07000	0.00000	-0.02926	1.14399	0.00000	-27.24968
-0.07000	0.00000	-0.02947	1.08155	0.00000	-27.05905
-0.07000	0.00000	-0.02967	1.01858	0.00000	-26.87318
-0.07000	0.00000	-0.02987	-0.23534	-2.97600	-33.47679
-0.21400	0.05760	-0.03914	-6.38861	-3.18682	-33.42867
-0.31000	0.09600	-0.04870	-9.86613	-3.48737	-40.20672
-0.40600	0.13440	-0.06069	-13.11797	-3.91490	-48.11546
-0.50200	0.17280	-0.07618	-15.94567	-4.46799	-58.06284
-0.59800	0.21120	-0.09568	-18.26002	-5.14503	-69.71411
-0.69400	0.24960	-0.11965	-19.96976	-5.94426	-82.81303
-0.79000	0.28800	-0.14845	-20.99700	-6.86375	-97.10711
-0.88600	0.32640	-0.18234	-21.28271	-7.90150	-112.35280
-0.98200	0.36480	-0.22153	-20.79085	-9.05547	-128.31628
-1.07800	0.40320	-0.26615	-18.32043	-4.15460	-137.98941
-0.87550	0.38400	-0.24625	-8.24904	-2.45438	-158.12007
-0.77250	0.38400	-0.23570	-2.73634	-1.76805	-171.07873
-0.66950	0.38400	-0.22114	2.92372	-1.08650	-183.33646
-0.56650	0.38400	-0.20204	8.14367	-0.41035	-193.93938
-0.46350	0.38400	-0.17915	12.93039	0.26128	-203.12115
-0.36050	0.38400	-0.15305	17.27410	0.92909	-211.02662
-0.25750	0.38400	-0.12423	21.18317	1.59367	-217.79003

PAGE 1-2

TOUT

FRI 09-JUN-78 16:47

-0.15450	0.38400	-0.09310	24.67738	2.25550	-223.53195
-0.05150	0.38400	-0.05999	27.78449	2.91497	-228.36071
0.05150	0.38400	-0.02516	31.61952	0.92190	-226.20666
0.02625	0.33165	-0.03398	34.70709	0.73126	-229.72482
0.04375	0.29675	-0.02810	33.73462	0.79775	-227.20657
0.06125	0.26185	-0.02240	31.80390	0.84408	-223.25377
0.07875	0.22695	-0.01705	28.57249	0.87030	-217.26677
0.09625	0.19205	-0.01218	24.27926	0.87638	-209.52366
0.11375	0.15715	-0.00789	19.13745	0.86221	-200.23588
0.13125	0.12225	-0.00429	13.35612	0.82765	-189.59441
0.14875	0.08735	-0.00147	7.13494	0.77249	-177.77011
0.16625	0.05245	0.00050	0.66170	0.69650	-164.91599
0.18375	0.01755	0.00155	-5.88926	-0.48560	-151.16913
0.14875	-0.01735	-0.01089	-4.85386	-0.58869	-134.95767
0.13125	-0.05225	-0.01817	-6.04065	-0.66662	-119.19783
0.11375	-0.08715	-0.02482	-6.90771	-0.72437	-103.05099
0.09625	-0.12205	-0.03073	-7.33338	-0.76170	-86.60418
0.07875	-0.15695	-0.03583	-7.21623	-0.77833	-69.87564
0.06125	-0.19185	-0.04005	-6.45614	-0.77396	-52.87619
0.04375	-0.22675	-0.04333	-4.95428	-0.74825	-35.60723
0.02625	-0.26165	-0.04561	-2.61187	-0.70086	-18.06194
0.00875	-0.29655	-0.04683	0.67068	-0.63140	-0.22595
-0.00875	-0.33145	-0.04696	2.83091	0.00300	5.58817
-0.00000	-0.26165	-0.04690	1.96415	-0.03804	24.11995
-0.00000	-0.22675	-0.04684	3.16809	-0.03292	34.02487
-0.00000	-0.19185	-0.04677	4.44108	-0.02781	42.51945
-0.00000	-0.15695	-0.04670	5.44271	-0.02272	48.36358
-0.00000	-0.12205	-0.04663	6.06160	-0.01764	51.81780
-0.00000	-0.08715	-0.04657	6.24855	-0.01258	53.05794
-0.00000	-0.05225	-0.04650	6.00990	-0.00753	52.24964
-0.00000	-0.01735	-0.04643	5.39385	-0.00250	49.54589
-0.00000	0.01755	-0.04636	4.47974	0.00252	45.08815
-0.00000	0.05245	-0.04630	4.45131	0.00753	45.17400
-0.00000	0.03500	-0.04623	3.37520	0.00502	37.57075
-0.00000	0.03500	-0.04616	2.77290	0.00501	33.27076
-0.00000	0.03500	-0.04610	2.21299	0.00500	28.65718
-0.00000	0.03500	-0.04603	1.75466	0.00499	24.43748
-0.00000	0.03500	-0.04597	1.37674	0.00499	20.54844
-0.00000	0.03500	-0.04590	1.06443	0.00498	16.96583
-0.00000	0.03500	-0.04584	0.80529	0.00497	13.66563
-0.00000	0.03500	-0.04577	0.58897	0.00497	10.62574
-0.00000	0.03500	-0.04571	0.40689	0.00496	7.82576
-0.00000	0.03500	-0.04564	0.25195	0.00495	5.24686
-0.00000	0.03500	-0.04558	0.11827	0.00495	2.87165
-0.00000	0.03500	-0.04551	0.00105	0.00494	0.68410
-0.00000	0.03500	-0.04545	-0.10369	0.00493	-1.33059
-0.00000	0.03500	-0.04538	-0.19916	0.00492	-3.18608
-0.00000	0.03500	-0.04532	-0.28800	0.00492	-4.89497
-0.00000	0.03500	-0.04526	-0.37232	0.00491	-6.46886
-0.00000	0.03500	-0.04519	-0.45382	0.00490	-7.91847
-0.00000	0.03500	-0.04513	-0.53387	0.00490	-9.25368
-0.00000	0.03500	-0.04507	-0.61352	0.00489	-10.48359
-0.00000	0.03500	-0.04500	-0.69361	0.00488	-11.61660
-0.00000	0.03500	-0.04494	-0.77477	0.00488	-12.66045
-0.00000	0.03500	-0.04488	-0.85749	0.00487	-13.62226
-0.00000	0.03500	-0.04482	-0.94212	0.00486	-14.50861
-0.00000	0.03500	-0.04475	-1.02889	0.00486	-15.32555
-0.00000	0.03500	-0.04469	-1.11797	0.00485	-16.07865

PAGE 1-3

TOUT

FRI 09-JUN-78 16:47

-0.00000	0.03500	-0.04463	-1.20945	0.00484	-16.77304
-0.00000	0.03500	-0.04457	-1.30337	0.00484	-17.41347
-0.00000	0.03500	-0.04451	-1.39972	0.00483	-18.00428
-0.00000	0.03500	-0.04444	-1.49847	0.00482	-18.54949
-0.00000	0.03500	-0.04438	-1.59955	0.00482	-19.05278
-0.00000	0.03500	-0.04432	-1.70288	0.00481	-19.51755
-0.00000	0.03500	-0.04426	-1.80837	0.00480	-19.94694
-0.00000	0.03500	-0.04420	-1.91591	0.00480	-20.34380
-0.00000	0.03500	-0.04414	-2.02540	0.00479	-20.71081
-0.00000	0.03500	-0.04408	-2.13673	0.00478	-21.05038
-0.00000	0.03500	-0.04402	-2.24978	0.00478	-21.36475
-0.00000	0.03500	-0.04396	-2.36444	0.00477	-21.65600
-0.00000	0.03500	-0.04390	-2.48061	0.00476	-21.92601
-0.00000	0.03500	-0.04384	-2.59816	0.00476	-22.17653
-0.00000	0.03500	-0.04378	-2.71701	0.00475	-22.40915
-0.00000	0.03500	-0.04372	-2.83704	0.00474	-22.62535
-0.00000	0.03500	-0.04366	-2.95818	0.00474	-22.82648
-0.00000	0.03500	-0.04360	-3.08032	0.00473	-23.01378
-0.00000	0.03500	-0.04354	-3.20338	0.00472	-23.18840
-0.00000	0.03500	-0.04348	-3.32728	0.00472	-23.35138
-0.00000	0.03500	-0.04343	-3.45195	0.00471	-23.50370
-0.00000	0.03500	-0.04337	-3.57731	0.00471	-23.64622
-0.00000	0.03500	-0.04331	-3.70331	0.00470	-23.77978
-0.00000	0.03500	-0.04325	-3.82987	0.00469	-23.90511
-0.00000	0.03500	-0.04319	-3.95694	0.00469	-24.02290
-0.00000	0.03500	-0.04313	-4.08447	0.00468	-24.13377
-0.00000	0.03500	-0.04308	-4.21240	0.00467	-24.23832
-0.00000	0.03500	-0.04302	-4.34070	0.00467	-24.33705
-0.00000	0.03500	-0.04296	-4.46932	0.00466	-24.43047
-0.00000	0.03500	-0.04290	-4.59821	0.00466	-24.51902
-0.00000	0.03500	-0.04285	-4.72735	0.00465	-24.60311
-0.00000	0.03500	-0.04279	-4.85670	0.00464	-24.68310
-0.00000	0.03500	-0.04273	-4.98622	0.00464	-24.75936
-0.00000	0.03500	-0.04268	-5.11590	0.00463	-24.83218
-0.00000	0.03500	-0.04262	-5.24569	0.00462	-24.90187
-0.00000	0.03500	-0.04256	-5.37559	0.00462	-24.96869

PAGE 1-1

DINPUT

MON 31-JUL-78 13:05

24 - NDEV

**roll-away
constants,
subliminal
washout**

0.0025

0.2000

1.0000

0.6500

0.6500

TMAX, DT
2660
0.2500

00 - 5
00 0.
0000

0.100
1.570
0 0.0

5.0000
0.0500
0.0000

3

roll-sway
constants,
linear
washout

DT
MAX, DT
2660

0000

0.100
1.570

- NDEV
5.0000
0.0500

24-150

	SUBLIMINAL	LINEAR	
PDLA	0.20E+03	0.20E+03	} J_p measures
PDL1	1.03192	1.03192	
PDL2	0.63262	0.63262	
PDL3	0.78101	0.78101	
PDL4	0.33483	0.33483	
PDL5	0.22039	0.22039	
PSENA	0.16E+03	0.16E+03	
PSEN1	1.02612	1.02612	
PSEN2	0.75415	0.75415	
PSEN3	0.72697	0.72697	
PSEN4	0.40287	0.40287	
PSENS	0.20456	0.20456	
PNLA	0.13E+03	0.13E+03	
PNL1	0.96801	0.96801	
PNL2	0.04672	0.04672	
PNL3	0.98307	0.98307	
PNL4	0.02354	0.02354	
PNL5	0.00048	0.00048	
GS1	0.00000	0.00000	
GS2	0.00000	0.00000	
GS3	0.00000	0.00000	
PACC	0.00000	0.00000	
PVEL	30.61460	30.61460	
PDIS	29.79046	29.79046	

PDLA	0.00266	0.00266	} J_{yT} measures
PDL1	0.68968	0.68925	
PDL2	0.00173	0.00081	
PDL3	0.84229	0.80090	
PDL4	-0.02917	-0.03374	
PDL5	0.00016	0.00010	
PSENA	0.00100	0.00100	
PSEN1	0.55926	0.00630	
PSEN2	1.03190	0.01605	
PSEN3	0.36271	0.03645	
PSEN4	-0.00837	0.01462	
PSENS	0.78E-04	0.37E-04	
PNLA	0.45E-04	0.45E-04	
PNL1	0.96575	0.92194	
PNL2	-0.00272	0.00253	
PNL3	0.98200	0.96018	
PNL4	-0.00185	0.00132	
PNL5	0.13E-03	0.27E-03	
GS1	0.00000	0.00000	
GS2	0.00000	0.00000	
GS3	0.00000	0.00000	
PACC	27.97919	36.29805	
PVEL	16.43408	24.66418	
PDIS	21.68047	52.06570	

PDLA	0.00251	0.00251	} J_y measures
PDL1	0.38290	0.00414	
PDL2	0.59564	0.00901	
PDL3	0.24131	0.02167	
PDL4	-0.06870	0.00157	

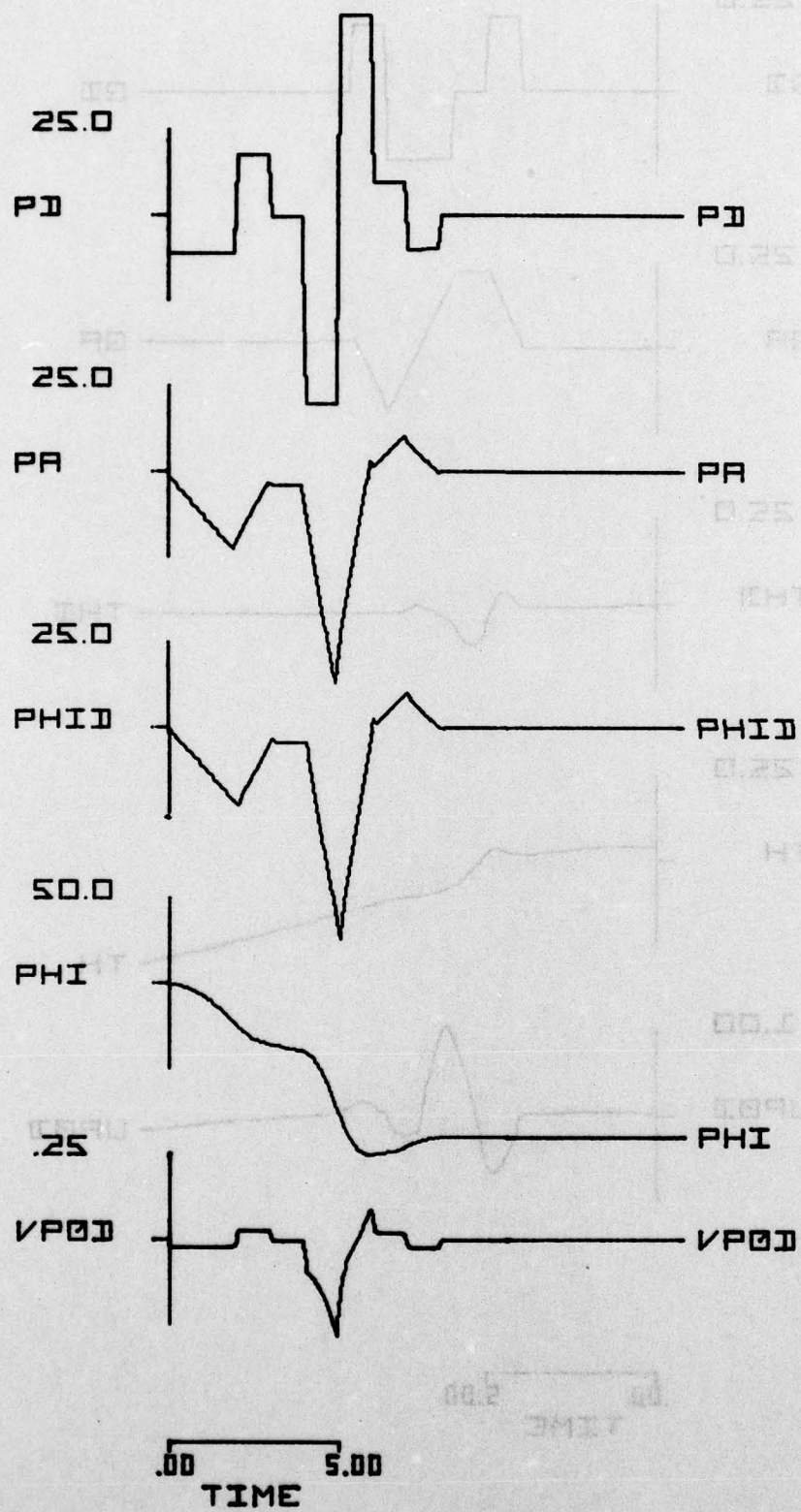
MON 31-JUL-78 13:03

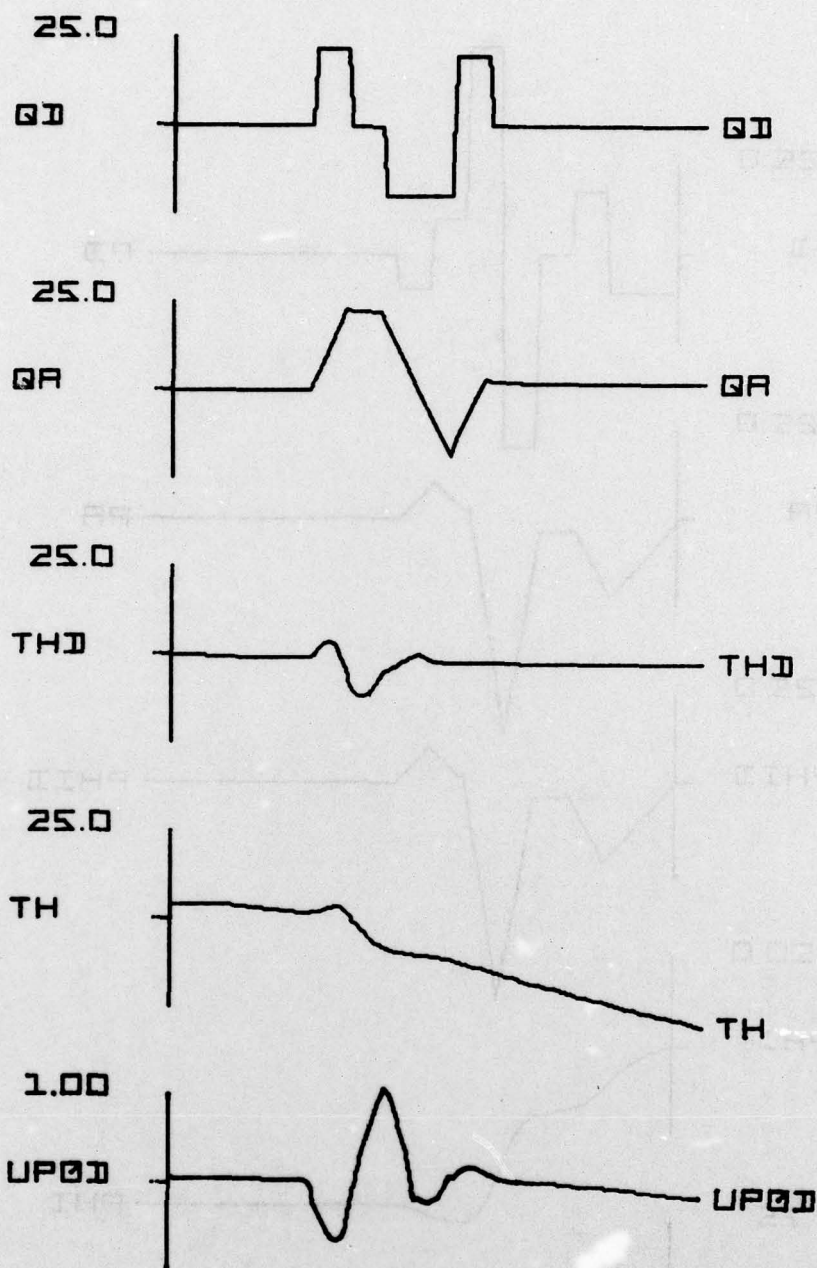
PAGE 1-1

SWAY

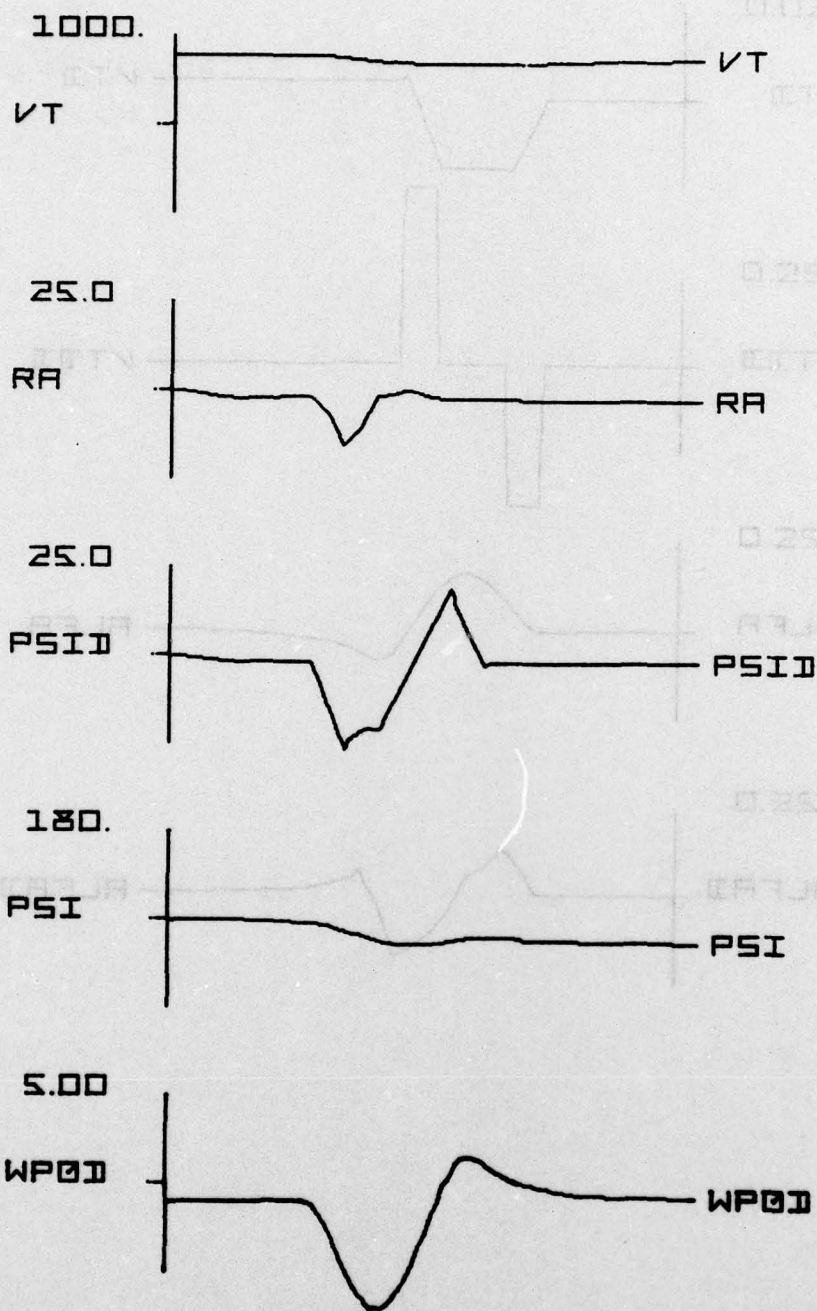
MON 31-JUL-78 13:04

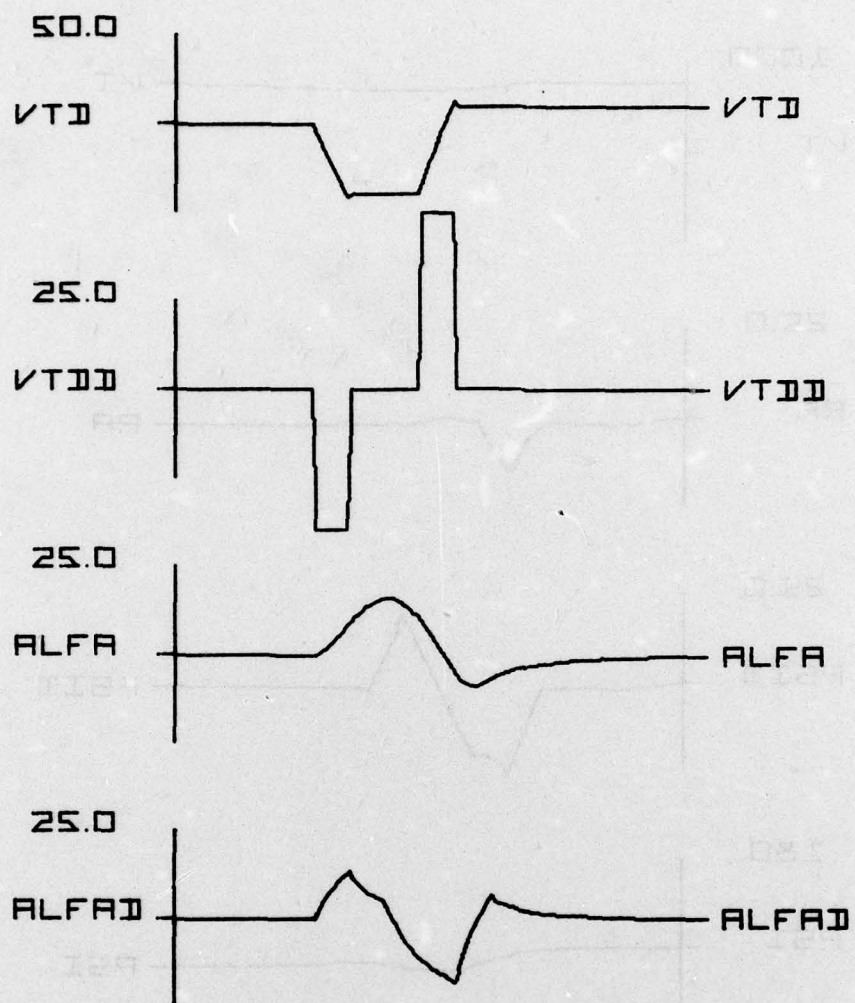
0.00	0.00
0.69	0.61
0.00	0.00
0.84	0.80
-0.03	-0.03
0.00	0.00
0.00	0.00
0.56	0.01
1.08	0.02
0.36	0.04
-0.01	0.01
0.00	0.00
0.00	0.00
0.97	0.92
-0.00	0.00
0.98	0.96
-0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
27.98	36.30
16.43	24.66
21.68	52.07
0.00	0.00
0.38	0.00
0.60	0.01
0.24	0.02
-0.07	0.00





.00 5.00
TIME





.00 5.00
TIME

DISCRETE DATA FOR REMAINING SCENARIOS

Scenario 1, HQDT1: 5 discrete data points

2.0000	0.0000	0.0000	0.0000
3.0000	0.4400	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000
6.0000	-0.4400	0.0000	0.0000
11.0000	0.0000	0.0000	0.0000

Scenario 2, HQDT2: 9 discrete data points

2.0000	0.0000	0.0000	0.0000
3.0000	0.4400	0.0000	0.0000
5.0000	0.0000	0.0000	0.0000
6.0000	-0.4400	0.0000	0.0000
8.0000	0.0000	0.0000	0.0000
9.0000	-0.6600	0.0000	0.0000
12.0000	0.0000	0.0000	0.0000
13.0000	0.6600	0.0000	0.0000
17.0000	0.0000	0.0000	0.0000

Scenario 4, A/G: 40 discrete data points

3.0000	0.0000	0.0000	0.0000
5.0000	0.0000	0.0000	2.5000
6.0000	-0.5200	0.0350	0.0000
7.0000	0.0000	0.0350	0.0000
8.0000	0.5200	0.0350	0.0000
22.4000	0.0000	0.0000	0.0000
23.4000	0.5200	-0.0350	0.0000
24.4000	0.0000	-0.0350	0.0000
25.4000	-0.5200	-0.0350	0.0000
45.0000	0.0000	0.0000	0.0000
46.0000	-0.8700	0.0750	4.0000
47.0000	0.0000	0.0750	4.0000
48.0000	0.8700	0.0750	4.0000
52.0000	0.0000	0.0000	0.0000
53.0000	0.8700	-0.0750	0.0000
54.0000	0.0000	-0.0750	0.0000
55.0000	-0.8700	-0.0750	0.0000
65.5000	0.0000	0.0000	0.0000
66.5000	0.0000	0.1640	-4.0000
67.0000	0.0000	0.0000	-4.0000
68.0000	-0.7000	0.0070	-4.0000
68.5000	0.0000	0.0070	-4.0000
69.0000	0.0000	0.0070	-2.5000
70.0000	0.7000	0.0070	-2.5000
70.5000	0.0000	0.0070	-2.5000

DISCRETE DATA FOR REMAINING SCENARIOS

Scenario 4 (cont'd)

73.5000	0.0000	0.0070	0.0000
74.0000	0.0000	0.0000	0.0000
74.5000	0.0000	-0.0600	0.0000
75.5000	0.7000	-0.0600	0.0000
76.5000	0.0000	-0.0600	0.0000
77.5000	-0.7000	-0.0600	0.0000
93.0000	0.0000	0.0000	0.0000
94.0000	-0.5200	0.0300	0.0000
95.0000	0.0000	0.0300	0.0000
96.0000	0.5200	0.0300	0.0000
110.0000	0.0000	0.0000	0.0000
111.0000	0.5200	-0.0300	0.0000
112.0000	0.0000	-0.0300	0.0000
113.0000	-0.5200	-0.0300	0.0000
120.0000	0.0000	0.0000	0.0000

Scenario 5, AA1: 42 discrete data points

2.0000	0.0000	0.0000	0.0000
3.0000	0.4360	0.0000	0.0000
4.0000	0.0000	0.0000	0.0000
5.0000	-0.4360	0.0000	0.0000
6.0000	0.0000	0.0000	0.0000
8.0000	0.0000	0.0610	0.0000
9.0000	0.0000	0.0000	0.0000
10.0000	0.0000	0.0000	-10.0000
11.0000	0.1130	0.0000	-10.0000
12.0000	0.1130	0.0000	0.0000
13.0000	0.0000	0.0000	0.0000
14.0000	-0.1130	0.0000	0.0000
15.0000	-0.1130	-0.0310	0.0000
16.0000	-0.0700	-0.0310	10.0000
17.0000	0.0000	0.0000	10.0000
21.0000	0.0000	0.0000	0.0000
23.0000	0.0000	0.0000	2.0000
29.0000	0.0000	0.0000	0.0000
30.0000	0.0700	0.0000	0.0000
31.0000	0.0000	0.0000	0.0000
32.0000	0.3320	0.0000	0.0000
33.0000	0.0000	0.0000	0.0000
34.0000	0.0000	0.0700	0.0000
35.0000	-0.3320	0.0000	0.0000
36.0000	-0.0870	0.0000	0.0000
37.0000	0.0000	-0.0700	0.0000
45.0000	0.0000	0.0000	0.0000
46.0000	0.0870	0.0000	0.0000
47.0000	0.3140	0.0000	0.0000
48.0000	0.0000	0.0000	0.0000
49.0000	-0.3140	0.0000	0.0000

DISCRETE DATA FOR REMAINING SCENARIOS

Scenario 5 (cont'd)

51.0000	0.0000	0.0000	0.0000
52.0000	0.0000	0.0000	-2.0000
53.0000	0.0000	0.0700	-2.0000
54.0000	0.0000	0.0000	0.0000
57.0000	0.3140	0.0000	0.0000
58.0000	-0.3140	0.0000	0.0000
59.0000	-0.3490	0.0000	0.0000
60.0000	0.1220	-0.0350	0.0000
62.0000	0.1220	0.0000	0.0000
63.0000	0.0000	0.1400	0.0000
65.0000	0.0000	0.0000	0.0000

Scenario 6, AA2: 16 discrete data points

1.0000	0.0000	0.0000	0.0000
2.0000	0.5410	0.0000	0.0000
2.5000	0.5410	0.0350	0.0000
4.0000	-0.5410	0.0350	0.0000
6.0000	-0.1400	0.0350	0.0000
8.0000	0.1400	0.0350	0.0000
9.0000	0.3500	0.0350	-24.3000
10.0000	0.0520	0.0350	-24.3000
12.0000	0.0000	0.0350	0.0000
13.0000	-0.4010	0.0000	0.0000
14.0000	-3.7520	0.0000	0.0000
14.5000	0.0000	-3.6650	0.0000
15.0000	0.0000	3.6650	0.0000
16.0000	3.7520	0.0000	24.3000
17.0000	1.0470	0.0000	24.3000
18.0000	0.0000	0.0000	0.0000

Scenario 7, AA3: 8 discrete data points

3.0000	0.0000	0.0000	0.0000
4.0000	0.5240	0.0210	0.0000
5.0000	0.0000	0.0210	0.0000
6.0000	-0.5240	0.0210	0.0000
7.0000	-0.9600	0.0210	0.0000
8.0000	0.0000	0.0210	-25.0000
9.0000	0.9600	0.3190	-25.0000
11.0000	0.0000	0.0000	0.0000

APPENDIX A
NUMERICAL INTEGRATION TECHNIQUE

In general, numerical approximations are used whenever an integration is desired. The numerical technique used in this software is given below:

$$z(t) = z(t - \Delta t) + \Delta t/2 [3\dot{z}(t) - \dot{z}(t - \Delta t)]$$

where

$$\begin{aligned} z(t) &= \text{state at time } t \text{ sec} \\ z(t - \Delta t) &= \text{state at time } t - \Delta t \text{ sec} \\ \dot{z}(t) &= dz/dt \text{ at time } t \text{ sec} \\ \dot{z}(t - \Delta t) &= dz/dt \text{ at time } t - \Delta t \text{ sec} \\ \Delta t &= \text{integration interval (in sec)} \end{aligned}$$

For these subroutines, an integration interval of 0.1 seconds was used ($\Delta t = 0.1$).

This method is implemented within the subroutines by saving the previous values of $z(t)$ and $\dot{z}(t)$ in the arrays $z(t - \Delta t)$ and $\dot{z}(t - \Delta t)$. In FORTRAN variable name notation, the current values of the states and their derivatives are stored in the Z and ZD arrays, while the past values (at $t - \Delta t$) of the states and their derivatives are stored in the ZO and ZDO arrays. Thus, the FORTRAN expression which implements this numerical approximation is:

$$Z = ZO + DELT2*(3.0*ZD - ZDO)$$

REFERENCE

1. Hildebrand, F. B., Advanced Calculus for Applications, Prentice Hall, Englewood Cliffs, N.J., 1976, pp. 96-102.

APPENDIX B

NOTES ON SOFTWARE USE

1. To execute the scenario generation package:
 - a. Place appropriate PROGRAM card in front of NEWMAN mainline routine.
 - b. The following routines must be linked to execute scenario operation:

mainline	(File name NEWMAN)	
SENGEN	}	(File name SENLIB)
DATGEN		
READAT		
HQDT4		
ERRMES	}	(File name PLTLIB)
OUTR		
GPDPLT		

2. To execute drive logic simulation/measure generation package:
 - a. Place appropriate PROGRAM card in front of NEWMAN mainline routine. This PROGRAM card should contain the name of each tape to be used; since each set of scenario data resides on a separate tape (conceptually, at least), a tape name should appear on the PROGRAM card for each set of scenario data to be processed through the drive logic.
 - b. The following routines must be linked to execute drive logic — sensory apparatus simulation

ROLSWA	(File name ROLSWA)	
or PITSUR	(File name PITSUR)	
or YAWHEV	(File name YAWHEV)	
THRES	}	(File name DLLIB, only necessary if ROLSWA is used)
THRESD		